AI-Driven efficiency: Large Language Models in Business Process Optimization

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Business Informatics

eingereicht von

Thomas Laner, BSc
Matrikelnummer 11807845

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.-Prof. Mag. Dr. Walter Schwaiger, MBA

Wien, 0. 0

Thomas Laner

Walter Schwaiger

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Informatics

AI-Driven efficiency: Large Language Models in Business Process Optimization

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Thomas Laner, BSc
Registration Number 11807845

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. Mag. Dr. Walter Schwaiger, MBA

Vienna, 0, 0

Thomas Laner

Walter Schwaiger

Erklärung zur Verfassung der Arbeit

Thomas Laner, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 0. 0

Thomas Laner

Acknowledgements

Firstly, I would like to thank my supervisor, Univ.-Prof. Mag. Dr. Walter Schwaiger, MBA, for his guidance and support throughout this research. His feedback and direction were crucial in shaping the project, often providing valuable insights and new perspectives.

I am also grateful to the organization that provided the business process and BPMN models central to this study. Their cooperation gave this work real-world relevance, and without their contribution, it would not have been possible.

I would further like to acknowledge Andrej Karpathy for his tutorials on neural networks, which deepened my understanding of their inner workings and continue to shape how I interact with them.

Lastly, I want to thank my family for their unwavering support throughout both of my degrees. Their encouragement has been invaluable, and I could not have made it without them.

Kurzfassung

Die Effizienz und Anpassungsfähigkeit eines Unternehmens an wandelnde Marktanforderungen und technologischen Fortschritt werden maßgeblich durch dessen Geschäftsprozesse determiniert. Das Management dieser Prozesse, stützt sich hierbei traditionell auf menschliches Fachwissen, insbesondere bei der Geschäftsprozessoptimierung (GPO), welche die Analyse und Verbesserung dieser Prozesse umfasst. Angesichts der zunehmenden Komplexität dieser Prozesse und des wachsenden Fachkräftemangels sehen sich Unternehmen jedoch vermehrt mit einer Ressourcenlücke konfrontiert, welche nicht durch gegenwärtig verfügbare Hilfsmittel geschlossen werden kann.

Eine vielversprechende Lösung dieses Problems, wird nun von aktuellen Machine Learning Modellen, den sogenannten Large Language Models (LLMs), geboten. Diese sind in der Lage, kontextrelevante Informationen zu generieren, und somit menschliche Entscheidungsprozesse nachzuahmen. Da das Potenzial von LLMs zur Automatisierung der GPO allerdings noch weitgehend unerforscht ist, zielt diese Arbeit darauf ab, diese Forschungslücke zu schließen. Dabei werden wir von folgender Frage geleitet: *Können Large Language Models genutzt werden, um Geschäftsprozessoptimierung zu automatisieren und dadurch die Einschränkungen der aktuellen Automatisierungsmethoden zu überwinden?*

Die auf diese Weise entstandene Arbeit basiert auf einem Action Research Ansatz, der sich über fünf iterative Zyklen erstreckt. In den ersten beiden Zyklen erfolgt eine Untersuchung der syntaktischen Fähigkeiten von LLMs im Hinblick auf die Einhaltung von Prozessmodellierungskriterien, während die folgenden Zyklen sich den semantischen Fähigkeiten widmen, welche das Erkennen und Lösen von Optimierungspotenzialen zum Gegenstand haben.

Unsere Ergebnisse zeigen, dass aktuelle state-of-the-art LLMs bestimmte Aspekte der GPO automatisieren können, jedoch weiterhin menschliche Expertise bei Prozessen, die Abhängigkeiten zwischen Prozessschritten und komplexe Abläufe beinhalten, erforderlich ist. Der stake Einfluss, welchen Kontext und explizite Lenkung hierbei auf die Ergebnisse haben, sowie damit verbundene Herausforderungen wie Overfitting unterstreichen hierbei die Wichtigkeit ein sorgfältiges Vorgehens. Die Ergebnisse unserer Arbeit legen nahe, dass eine vollständige Automatisierung entweder von Weiterentwicklungen der LLMs, einer tiefen Integration in die Systeme der jeweiligen Organisation, um den Modellen ausreichenden Kontext bereitzustellen, oder von einer gezielten Vorgehensweise und einem durchdachten Prompt-Design, oder einer Kombination dieser Ansätze, abhängig ist.

Abstract

The efficiency and adaptability of an organization are significantly influenced by its business processes, which directly determine its ability to align with evolving market demands and technological advancements. Business Process Management (BPM) has traditionally relied on human expertise, particularly in the context of Business Process Optimization (BPO), which involves the analysis and redesign of processes with the objective of eliminating inefficiencies. As processes become increasingly complex and the shortage of skilled workers persists, organizations are confronted with a widening resource gap that current tools are unable to resolve, as they are incapable of automating optimization.

A promising solution to this issue is offered by novel machine learning models, the so-called Large Language Models (LLMs), which are capable of mimicking human decision-making processes by generating contextually relevant insights. Nevertheless, their capacity to automate BPO remains largely under-explored, representing a notable research gap. This work is therefore guided by the question: *Can Large Language Models be leveraged to automate BPO, thereby overcoming the limitations of current automation methods?*

The study answers this question in five iterative cycles by leveraging an action research framework. Thereby, the initial two cycles are focused on evaluating the syntactic capabilities of LLMs in adhering to process modeling criteria. The subsequent cycles are then focused on evaluating their semantic ability to identify inefficiencies and implement process optimizations.

Our results indicate that, while current state-of-the-art LLMs are capable of automating certain aspects of BPO, they still require human supervision for complex workflows involving task interdependencies and intricate process flows. The strong influence of context and explicit guidance on model performance, along with the associated challenges of overfitting, demand careful management. This suggests that full automation will depend on further model advancements, deeper integration into organizational systems, or a more carefully crafted approach and prompt design.

Contents

Kurzfassung **ix**

Abstract **xi**

1 Introduction **1**
1.1 Motivation 1
1.2 Identified Gap and resulting Objectives of this work 2
1.3 Research Strategy 4
1.4 Structure of the Thesis 5

2 Related Work **7**
2.1 Business Process Optimization 7
2.2 Large Language Models 10
2.3 Application of LLMs for BPM and BPO 12
2.4 Discussion 14

3 Methodological Framework **17**
3.1 Action Research 17
3.2 Rationale for adopting the AR-Methodology 18
3.3 Application of the Methodology in this work 19

4 Requirements Assessment and Decision Rationale **23**
4.1 Business Process 23
4.2 Large Language Model 26
4.3 Evaluation 28

5 Adherence to Modeling Criteria **33**
5.1 Assessment of Process Model Quality 33
5.2 Context-Enhanced Assessment of Process Model Quality 38

6 Process Optimization Capabilities **43**
6.1 Assessment of Process Optimization Capabilities 43
6.2 Assessment of Process Optimization Capabilities with adapted input method 47

6.3 Context-Enhanced Assessment of Process Optimization Capabilities . 53

7 Discussion 61
7.1 Analysis of Results 61
7.2 Relation and Contributions to Existing Literature 63
7.3 Limitations of our work 65
7.4 Future Research Directions 66

8 Conclusion 67
8.1 Summary of Findings 67
8.2 Implications to the Field 67
8.3 Final Thoughts 68

9 Appendix 69
9.1 Appendix to Chapter 5 70
9.2 Appendix to Chapter 6 85

Overview of Generative AI Tools Used 135

List of Figures 137

List of Tables 139

Acronyms 141

Bibliography 143

CHAPTER 1

Introduction

1.1 Motivation

In today's rapidly evolving business environment, which is characterized by intensified competition, globalization, and constant technological advancement, organizations are under constant pressure to enhance efficiency, quality, and adaptability in order to maintain their competitive position [MLB$^+$16, Alt23]. As a result, the importance of Business Process Management (BPM) has become increasingly evident as it offers a structured framework for the management and optimization of business processes, ensuring their alignment with organizational goals and the delivery of optimal value [KHH18]. By systematically analyzing, designing, implementing, monitoring, and optimizing processes, BPM enables organizations to streamline operations, reduce costs, and improve overall performance [RvB15, DRMR13].

With digitization and rapid innovation transforming industries, businesses must adopt an agile and responsive BPM framework to stay competitive in the digital age [AVL20]. This demands a State of the Art (SOTA) BPM framework, capable of continuous process optimization, with BPM's process optimization aspect, Business Process Optimization (BPO), having to be regarded as an integral aspect of BPM [Kir22].

Through these increased demands and an ever increasing shortage of skilled workers, manual and solely human reliant process management methods have become increasingly impractical and costly [For24, AVL20]. In his survey of the current state of BPM, Van der Aalst highlights a gap of tools in the area of process optimization, concluding that there is a notable lack of tools that suggest optimizations of processes [Aal13].

Current systems such as Robotic Process Automation (RPA), Process Mining (PM), Business Process Management Suites (BPMS), and classical machine learning (CML), as shown in the timeline in Figure 1.1, primarily focus on performance analysis and monitoring, leaving optimization to manual efforts rather than offering automated suggestions or improvements [WLC17, MFK$^+$21, SBW24, AAAHABH21].

Thus, current BPO automation tools are primarily of supportive nature, offering only assistance in monitoring and analysis, leaving the critical task of process optimization to human decision-makers. As a result, they are unable to close the resource gap outlined above, hindering organizations from achieving the operational efficiency and responsiveness required in today's fast-paced business landscape.

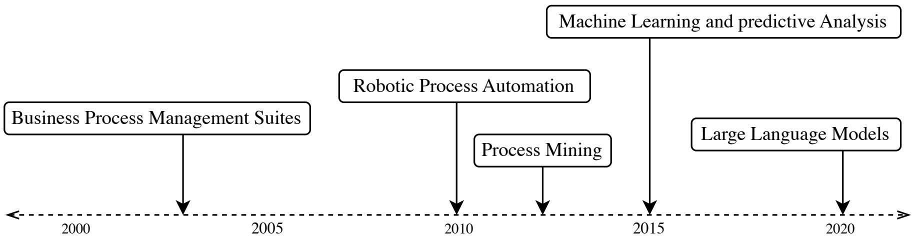
Machine Learning and predictive Analysis
Robotic Process Automation
Business Process Management Suites
Process Mining
Large Language Models
2000
2005
2010
2015
2020

Figure 1.1: Timeline of BPM Automation Tool Emergence

Large Language Models (LLMs), such as OpenAI's Generative Pre-Trained Transformer (GPT)-models, that gained worldwide renown through the *ChatGPT*-platform, represent a significant advancement in Deep Learning and entail significant disruptive potential [Ope24a, Tim24]. LLMs are trained on vast amounts of text data, enabling them to recognize patterns, generate context-appropriate human-like data, and thus mimic human decision making processes [Ope18, BGMMS21]. Unlike traditional Machine Learning methods, LLMs can process and analyze complex, unstructured data.

One of the first applications that gained renown outside of the *ChatGPT* has been *BloombergGPT*, a LLM designed for the financial industry, trained on a mixed dataset of financial and general data [WIL$^+$23]. It achieved best-in-class performance on financial tasks while maintaining competitive results on general-purpose benchmarks. Further notable use cases are their proven ability outperform financial analysts [KMN24], autonomously learn [WXJ$^+$23], as well as their ability to act as optimizers across various domains [YWL$^+$24].

These capabilities make LLMs well-suited for addressing the limitations of current BPO tools, offering significant potential for more adaptive, scalable, and efficient process management [ETdROR24, VBM23a].

1.2 Identified Gap and resulting Objectives of this work

The research on integrating LLMs into BPM is still in its early stages, demonstrated by the fact that many studies on the topic are currently undergoing publication [ETdROR24]. Nevertheless, due to rapidly increasing model performance, the adoption of LLMs across different fields, including BPM, is steadily accelerating [ETdROR24].

A review of the scientific literature on the applications of LLMs for BPM-Lifecycle tasks, with a particular focus on BPO (see section 2.3), revealed that only a limited number of phases are currently covered by SOTA research. These are primarily the discovery, analysis and monitoring phases, while the redesign phase remains particularly

understudied (see Figure 1.2). This gap is particularly pressing when considering the automation of BPO, as the redesign phase is where new process models are developed based on the preceding model and the identified issues.
Torres et al. reached a similar conclusion in their analysis of the existing literature, noting that the lifecycle phases currently being addressed are closely tied to text processing and information extraction [ETdROR24].
This identified gap has led several researchers to call for the exploration of the potential integration of LLMs into the remaining lifecycle phases [VBM23a, ETdROR24]. For further details on the reviewed literature, see chapter 2.

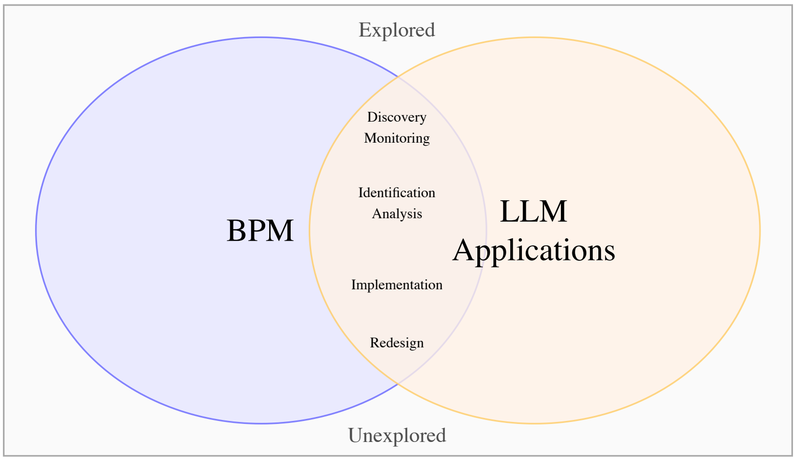

Figure 1.2: State of research for LLM-driven BPM

The problem this work aims to solve is to address the research gap in the SOTA literature regarding the feasibility of applying LLMs to automate BPO. Specifically, this work will focus on evaluating the capabilities of LLMs for BPO as a whole, particularly in integrating the analysis phase and the currently understudied redesign phase. The goal is to provide a more comprehensive understanding of LLMs's potential for automating BPO and to demonstrate their ability to reduce reliance on human expertise, thereby decreasing the significant resource demands of traditional approaches and beginning to close the outlined resource gap.

The research question which guides our work in closing this gap, is thus formulated as:

ORQ *Can Large Language Models be leveraged to automate business process optimization, thereby overcoming the limitations of current automation methods?*

1.3 Research Strategy

The Methodological Approach employed in this work is rooted in the Action Research (AR) framework (see chapter 3), which is particularly well-suited for investigating the capabilities of LLMs in dynamic environments such as BPO (see section 3.2). While traditional AR typically involves interventions with human practitioners, our study shifts the focus to the LLM itself, as we are concerned with its capabilities and behavior rather than those of its users. This places the LLM at the center of our investigation and aligns with other research that has already been conducted in the same area (e.g. [LZZ+24]). Given the human-like interaction configuration of LLMs, this methodology remains consistent with AR principles, while offering a novel adaptation tailored to the unique requirements of this emerging category of digital systems.

The Experimental Approach applied in this research is structured into five main action cycles, each building upon the insights gained from the previous one. The first two cycles, assess the syntactical capabilities of LLMs in the context of business processes, specifically evaluating their adherence to established business process modeling criteria. These cycles focus on the models' ability to detect violations of best practices and offer corrective suggestions. In the first cycle, simple prompts are used, while the second introduces additional context to examine how varying input depth impacts the models' outputs. The remaining three cycles shift the focus to the semantic capabilities of LLMs in process optimization, exploring their effectiveness in recognizing and solving optimization opportunities within processes. Hereby we leverage different input formats and different amounts of context, allowing for a more comprehensive understanding of LLM capabilities in the field.

The Scope and Limitations of this study are a consequence of several key decisions taken in order to guarantee practical relevance and enable the drawing of precise conclusions regarding the subject of our work. As outlined in chapter 4, the scope was intentionally narrowed to the application of a specific LLM (*GPT-4o*) using simple prompts, avoiding advanced prompting strategies, and focusing on a single process type (a digitization project process) while utilizing only two heuristics in the evaluation framework. This narrow focus was necessary to provide specific insights, but it introduces several limitations. The use of on a single process and it's subprocesses, as well as just two sets of heuristics for our evaluation, may not fully encompass the broader variability present in different domains or business processes. Furthermore, using only *GPT-4o* limits the ability to compare with alternative models. Finally, it is important to acknowledge the potential for biases inherent in the AR methodology and the hidden system prompts of the model to affect the findings. For a comprehensive dicsussion of these limitations and their implications, please refer to section 7.3.

1.4 Structure of the Thesis

This thesis is organized into eight chapters, starting with a review of the relevant literature and culminating in the application of methodology and evaluation of results. In this chapter the research motivation, the identified problem, and our research objectives and -strategy were introduced. Chapter 2 contains a review of the SOTA of the related scientific literature, leading to the formulation of our specific research questions that break the overarching research question down into smaller, more specific questions. Chapter 3 presents the AR methodology, discussing the rationale for its selection and its application in this study. Chapter 4 assesses the requirements that we pose to all components of our research, and outlines the evaluation framework that is applied in the action cycles. The AR cycles are detailed in the next two chapters. Chapter 5 focuses on the syntactic aspects of BPO, by evaluating LLM compliance with process modeling criteria, while chapter 6 focuses on the semantic capabilities of LLMs, by assessing the ability to recognize and solve optimization challenges in processes of varying complexity. Chapter 7 presents a discussion of the results, their relation to the existing literature, the limitations and biases of the study, as well as directions for further research. Finally, chapter 8 provides a summary of our work, discusses it's implications and offers concluding remarks.

CHAPTER 2

Related Work

In this chapter, we provide an overview of the SOTA of BPO and LLMs. Thereby, we first focus on the SOTA of BPO and LLMs separately and then on the work that has already been carried out in which LLMs have been leveraged to solve different tasks in the field of BPM, with a focus on BPO. Lastly, we discuss the discovered gaps and formulate specific research questions that break the overarching research question down into more manageable, focused inquiries.

2.1 Business Process Optimization

BPO is a practice in which businesses adapt their existing processes to new external and internal demands by making incremental improvements to them in order to realize the full potential of an existing process in a continuously changing environment [DRMR13].

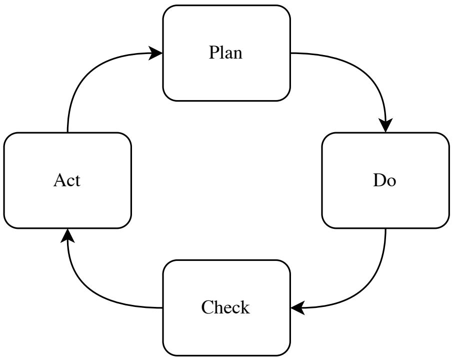

Figure 2.1: The Demming (PDCA) Cycle

This approach is closely aligned to the concept of the Deming cycle [Pet99] (see Figure 2.1), which focuses on continuous improvement through iterative cycles of planning, implementing, evaluating and refining.

Furthermore, the practice of BPO is part of Business Process Reengineering which differentiates between BPO and the more transformational and less commonly used Business Process Transformation in which a fundamental redesign of processes is performed [DRMR13]. The importance of these kinds of adaptations in today's rapidly changing environment cannot be understated as demonstrated by Liebeskind [Lie98] and Hungenberg et al. [HW98].

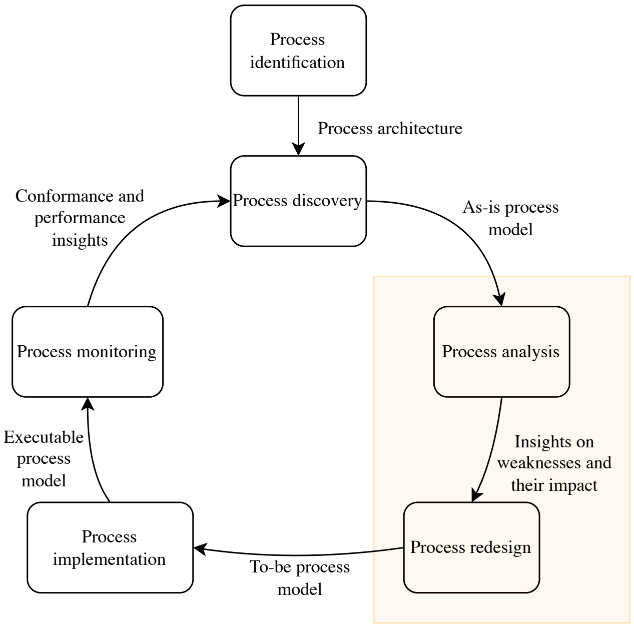

Figure 2.2: The BPM-Lifecycle and the focus of this work in it

Looking at Dumas' BPM-Lifecycle [DRMR13], which, in comparison to BPO encompasses the management of processes as a whole and not just their continued improvement, in Figure 2.2, BPO entails the process analysis phase in which a process is evaluated in order to determine inefficiencies, as well as the process redesign phase, in which a process is adapted in order to diminish or correct these inefficiencies. These phases are outlined in yellow in the figure above, demonstrating the focus of our work.

2.1.1 Current Efforts to Digitize BPO

As outlined in chapter 1, the increasing demands on BPO have led to more pronounced efforts to digitize the practice in recent years. As outlined above, traditional methods of BPO have relied heavily on techniques such as process redesign heuristics [Red05] and Six Sigma [Con10], which emphasize incremental improvements and quality control within existing frameworks. These methods often heavily involve manual analysis and deeply-driven insights, demanding expert-level knowledge to identify inefficiencies and propose optimizations.

Traditional BPO techniques face several limitations in today's dynamic and demanding business environment despite their widespread use. Dumas [DRMR13] notes that these manual methods are often time-consuming and resource-intensive, while Aalst [vdA11] highlights their reliance on expert knowledge, which can lead to inconsistencies in analysis and redesign due to human error and subjective judgment.

Current digitization tools for BPO such as the ones mentioned in section 1.1 still have quite a limited impact. Robotic Process Automation is effective at automating repetitive, rule-based tasks but falls short when it comes to analyzing and improving entire processes, leaving the actual optimization to human operators [WLC17]. Process Mining tools like Celonis [Cel24] are valuable for diagnosing inefficiencies by analyzing event logs, yet they primarily provide insights rather than direct solutions, requiring human intervention to interpret the data and implement improvements [MFK^{+}21]. Business Process Management Suite platforms such as ARIS [Sof24] offer frameworks for process management and execution but largely focus on providing a structured environment where processes can be modeled and monitored. The actual optimization still depends on manual efforts to refine and adapt these models [SBW24]. The gap in cognitive capabilities highlights the need for more advanced technologies that can enhance the depth and efficiency of process optimization efforts.
Machine learning and data analytics have been integrated into some BPO efforts to improve process insights and predictive capabilities. In this regard, the potential of machine learning in automating the reengineering of business processes to manage complexity and reduce implementation costs has been discussed by Angoudi et al. [AAAHABH21], while the concept of hybrid intelligence, the combination of human- and machine intelligence, as a means of improving BPM initiatives has been discussed by Aalst et al. [vdAW21]. In most of the current approaches however, the role of human expertise still remains essential as they are generally limited to structured data analysis, which requires high data quality, significant data preprocessing and extensive model training [NP20, Com21, BFI^{+}22].
All of these methods focus on streamlining repetitive tasks, improving data accuracy, and enhancing decision-making processes, however often fall short in handling complex, dynamic, and context-sensitive processes. Thus, while traditional BPO methods have served organizations well, the current digitization efforts reveal significant gaps that have

to be addressed in order to provide a more efficient and contemporary way of improving business processes.

2.2 Large Language Models

LLMs like OpenAI's GPT [Ope24b, Ope24a, Ope24c] or Meta's *Llama* [AI24b] have revolutionized the field of natural language processing (NLP) by enabling deep learning models to understand, generate, and interact with human language in increasingly sophisticated ways. The development of LLMs has been driven by advancements in neural network architectures, particularly the *transformer architecture* [VSP+17], which has become the foundation for most SOTA models today. Below an overview over the concepts that are covered in this section can be seen in Figure 2.3.

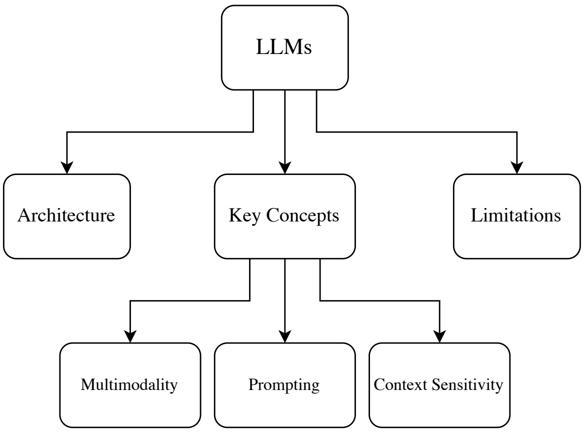

Figure 2.3: Overview over LLM aspects covered in this section

The Transformer Architecture was introduced by Vaswani et al. in 2017 [VSP+17]. It represents a significant breakthrough in NLP, as unlike previous models that relied on Recurrent Neural Networks [RHW86] and long short-term memory networks [HS97], the transformer utilizes self-attention mechanisms to process input data in parallel. This allows transformers to capture complex dependencies in text and significantly improves computational efficiency, making it the backbone for various LLMs, including OpenAi's GPT [Ope18].

Context Sensitivity in LLMs refers to the fact that the ability of models to understand and generate text is intrinsically influenced by the context in which the models calculate their output. The output of an LLM is determined not only by the immediate input, commonly referred to as the *prompt*, but also by the broader contextual information that shapes the model's perception of the task at hand. This broader context encompasses factors such as the *system prompt*, an underlying directive set by the developers and

typically concealed from end-users, and the preceding dialogue in a conversational thread [ZZPC23]. Furthermore, inherent model characteristics, such as training data, architecture, and parameters, also contribute to the contextual framework [ZMLA24, HC23].
Ensuring that LLMs are supplied with the appropriate contextual background is essential for optimizing their utility and precision. This exemplifies the necessity of aligning LLMs with human values and ethical standards, as context can substantially influence the ethical considerations and other aspects of their outputs [HC23, ZZPC23].

Prompting is a technique that guides LLMs by presenting them with specific contexts and questions, shaping the direction and quality of the output, thus allowing users to leverage the capabilities of LLMs more effectively [LXZ23]. Techniques such as *chain-of-thought* prompting, which encourages LLMs to generate intermediate reasoning steps, and *multi-agent based reasoning*, which employs interactions among multiple agents to simulate human-like dialogue and problem-solving processes, have been explored extensively [W^+22, LZY^+24]. Furthermore, in recent months, the so-called *prompt engineering*, which involves iterative refinement and experimentation to identify prompts that yield the most optimal responses from models has gained prominence [CZLZ24, Ope24d].
Despite these advancements many users do not fully exploit the potential of prompting [Cai24]. Studies indicate that while LLMs can perform a wide range of tasks, the effectiveness of their responses is often limited by suboptimal prompting practices [S^+22]. This gap underscores the need for greater education and tools that assist users in crafting effective prompts, thereby maximizing the benefits of LLM technologies across various applications.

Multimodality involves the integration of multiple data types, such as text- and image formats, to enhance the capabilities of AI models [Ope21]. Research on multimodal neurons highlights how models like CLIP can simultaneously process text and image data, reflecting a neural network's ability to understand complex content, similar to human cognition [Ope21]. The development of *GPT-4 Vision* further exemplifies the progress in this area, enabling models to analyze and interpret image inputs alongside text, thereby expanding the scope of tasks these models can perform [Ope23c]. Despite these advancements, challenges persist in fully leveraging the potential of multimodal systems, particularly regarding the integration and coordination of diverse data types [Ope23a].

Limitations are faced by LLMs despite and in some instances precisely because of their remarkable capabilities. One significant issue is hallucination, where models generate plausible-sounding but incorrect or nonsensical information [JLF^+23]. Hallucination arises because LLMs are trained on vast amounts of text data without an inherent understanding of the real world. They predict the next word or sequence based on learned statistical patterns rather than factual accuracy. This limitation can undermine the reliability of LLMs in critical applications, such as legal or medical advice, where

accuracy is essential.
Despite their vast training data and advanced capabilities in some areas, as demonstrated in chapter 1, LLMs have shown difficulties in handling seemingly easy tasks like negations and counting [WH24]. For instance, when asked to count objects or interpret statements involving negation, these models frequently produce incorrect outputs, reflecting their lack of true comprehension and logical reasoning abilities. This challenge further complicates their application in domains requiring precise and accurate information processing, such as technical writing or data analysis.
Moreover, currently LLMs require substantial computational resources for training and inference, which can limit their accessibility and environmental sustainability [BGMMS21]. The computational demands of LLMs are driven by their large number of parameters and the complexity of the training processes, which necessitate advanced hardware such as high-performance GPUs. In recent months, this caused significant increases in chip prices due to supply chain constraints and increased demand, exacerbating the cost of deploying LLMs and driving stock prices of high-end chip developers [New24d, New24b, CNB22, New24c]. In addition, the environmental impact of training LLMs is substantial. The energy consumption involved in training these models contributes to a larger carbon footprint, raising concerns about their sustainability in a world increasingly focused on reducing environmental impact [SGM19, New24a]. As such, the development of more efficient algorithms and hardware as well as the use of renewable energies for their training is essential to mitigate these issues and make LLMs more accessible and sustainable in the long term.

2.3 Application of LLMs for BPM and BPO

In order to tackle the issues that modern BPO-initiatives face (see section 2.1), researchers have begun to apply LLMs (see section 2.2) to different BPO-tasks. An overview of the existing body of work is provided by Bedilia et al. [ETdROR24], who have analyzed several studies that apply LLMs not only to BPO, but to other phases of the BPM lifecycle as well, with the aim of analyzing the current state of research in this area. They register several existing studies on the discovery, analysis and monitoring phase, but none on the other phases, thus also, crucially for our work, none on the redesign phase. Furthermore, they note that qualitative evaluation methods such as expert interviews are common in the studies that they analyzed, making their reproducibility difficult.

Process mining is an aspect of the process discovery phase that, according to our review, is already quite well covered by research. Grobs et al. demonstrate the capabilities of *GPT-4* for mining imperative and declarative process models from text sources as well as for the evaluation of process tasks for suitability for robotic process automation [GAER23]. It is worth noting that they achieved this without relying on specialized model configurations or prompt engineering, highlighting the potential advantages LLMs have over existing solutions.
Another paper that focuses on this aspect of BPM was written by Berti et al. [ber]. They

emphasize the need for benchmarks and evaluation strategies to improve performance, highlighting key capabilities such as handling long context windows and coding for text-to-SQL tasks, as well as addressing challenges such as hallucinations in LLM output.

The integration of LLMs into process modeling has been proposed by Kourani et al. with the goal of making process modeling more accessible to non-experts and more efficient by increasing the productivity of experts [KBSvdA24].
Their approach uses LLMs to automatically generate and refine process models from textual descriptions, providing a more intuitive and efficient approach compared to traditional methods. The framework uses prompting techniques and an interactive feedback loop to iteratively refine models.

Understanding Process-Oriented Modeling is a critical capability for models aiming to support users in task execution, as explored by Lins et al. [LNA$^+$23]. Their research investigates whether these models, such as *GPT-3.5*, can effectively interpret Business Process Model and Notation (BPMN) models from supplied *XML* [BPSM$^+$08] files. They find that while *GPT-3.5* can correctly initiate process instances, it encounters challenges in managing task dependencies and evaluating exclusive gateways.

Process Optimization on quantifiable processes has been explored by Li et al. [LMZ$^+$23] in developing *OptiGuide*, a framework that uses *GPT-4* to enhance supply chain optimization. Furthermore, they also developed an evaluation framework to assess LLM capabilities for BPM tasks. Their study demonstrates how LLMs can translate human queries into actionable optimization tasks, improving decision making and process efficiency.

Log Data processing which helps in monitoring of processes and the analysis of their performance has been tackled by Mudgal and Wouhaybi [MW23], who evaluated *ChatGPT*'s effectiveness in log parsing, analysis, and summarization. Their study found that while *ChatGPT* performs well in log parsing and identifying security information, it struggles with tasks like anomaly detection due to scalability limitations.

A specialized model is proposed by Kampik et al. [KWR$^+$23], who advocate for the creation and use of a *Large Process Model* rather than relying on general-purpose models. This model integrates generative AI with traditional symbolic and statistical methods to automate insights and actions in BPM using diverse data sources, promising significant industry impact and process improvement. However, it faces major feasibility challenges due to the extensive resources required.

Usage of LLMs in the analyzed works is quite interesting. Bedilia et al. noticed that of the 21 studies they reviewed, 17 use GPT-class models (7 use *GPT-4* and 5 use *GPT-3.5*), while other models such as Bard, *LLama2*, *PaLM 2*, and *Bert* were used only once each. They also note that only 2 studies apply fine-tuning while the others don't. GPT-class models have further been used by Grobs et al. [GAER23], Kourani et al. [KBSvdA24], Lins et al. [LNA$^+$23], Li et al. [LMZ$^+$23] and Mudgal and Wouhaybi [MW23]. The other studies did not mention which model they used or did not use one. Kourani et al. [KBSvdA24] used Alphabet's *Gemini* [AI24a] in addition to *GPT-4* [Ope23b].

2.4 Discussion

As we have seen in this chapter, modern requirements on BPO are significantly more demanding than they have been in the past. Traditional, manual methods of performing BPO, have become increasingly costly and inefficient due to the new found complexity and velocity which which organizations and their processes have to evolve.
As demonstrated by the literature, current attempts at automating BPO, mainly focus on aspects like monitoring and analyzing as-is process models, but not on a holistic automation of BPO. This means that the current methodologies are still heavily reliant on expert-driven insights, and are still subjected to the inherent limitations entailed by this approach. These insights, make it evident that a new and more disruptive approach is needed.

We believe, that we might have found this approach in using LLMs for BPO automation. These novel Deep Learning models offer unprecedented capabilities in generating context relevant text in human-like fashion and thus are already disrupting whole industries, as shown, for example, by *J.P. Morgan* in leveraging an LLM that can perform tasks traditionally handled by research analysts [Tim24]. Despite these exceptional capabilities, LLMs face some intricacies and limitations, that users and developers have to be mindful about.

As we have seen, other researchers have recognized the capabilities of these models for the field of BPM as well. Our analysis revealed that most of the papers in the area focus on singular BPM-Lifecycle steps or aspects of them with varying degree of coverage across different phases. Notably, process redesign is among the least represented steps. The only study we reviewed that addresses this phase is by Li et al. [LMZ$^+$23], which leverages quantitative data and thus relies on databases containing existing performance insights. This dependence on numerical data confines the approach to environments like supply chain management, where such data is abundant. However, this method may not be directly applicable to other BPM scenarios, as quantitatively assessing processes can pose a significant challenge if there is no direct quantitative information available for them. A holistic approach is taken only by Kampik et al. [KWR$^+$23], who, while proposing their Large Process Models, acknowledge the substantial resource requirements that an implementation would entail at the time, thus choosing not to implement it.
We justify the differing levels of coverage among the phases on the rationale that it is more challenging to assess the capabilities of LLMs for these phases than for other phases, such as process mining, where sufficient data is available on which to base an evaluation.

Regarding model usage, most of the papers we analyzed rely on generally available models, with the SOTA version of GPT at the time of writing being prominently featured. The authors cite widespread recognition, strong out-of-the-box performance, and an accessible user interface as the primary reasons for choosing generalist models. Recent research by Nori et al. supports this decision, showing that in the medical sector, generalist models like *GPT-4* can outperform specialized models that have undergone extensive domain-specific fine-tuning, provided effective prompt engineering techniques are employed [NLZ+23].

Thus, based on the literature, we were able to draw the following conclusions that influence our work:

1. **Heightened demands towards BPO:** Modern BPO has to handle more complex processes and shorter time-spans, making traditional methods inefficient.
2. **Limitations of Current Automation:** Current BPO automation focuses on monitoring and analyzing as-is processes but lacks holistic automation, remaining reliant on expert insights, making a more disruptive approach necessary to address modern BPO challenges.
3. **Potential of LLMs:** LLMs offer, despite current limitations, significant promise in BPO automation.
4. **Under exploration of LLM-capabilities for BPO:** Current BPM research under explores process redesign, with few studies addressing LLMs' potential for holistic, non-quantitative BPO due to evaluation complexity and resource demands.
5. **Prevalence of GPT Models in Research:** Most studies utilize GPT models, citing their strong performance and accessibility, with research showing generalist models can outperform specialized ones with effective prompting.

Based on these conclusions drawn from the literature, we have decided to focus our work on analyzing the current capabilities of SOTA, general-purpose LLMs in optimizing business processes, with emphasis on both the BPM analysis phase and the previously under-explored redesign phase. Our focus will be on transactional redesign approaches, as they are more common and easier to evaluate, as outlined above. From these decisions, we formulated several **specific research questions** to guide our work and break down our overarching research question.

SRQ1 *Are LLMs adhering to business process modeling criteria from the scientific literature?*

SRQ2 *Are LLMs correctly applying optimization criteria from the scientific literature when asked to optimize a process?*

SRQ3 *What specific limitations and challenges are identified in using LLMs for BPM-optimization tasks?*

CHAPTER 3

Methodological Framework

This chapter outlines the methodology chosen to evaluate the capabilities of LLMs in BPO. Our approach is grounded in the principles of AR, a reflective and iterative framework [BWH98] that aligns with the exploratory nature of working with LLMs.

3.1 Action Research

AR is a methodology characterized by its iterative cycles of planning, action, observation, and reflection [BWH98]. Originating from the work of Kurt Lewin in the 1940s, it was designed to facilitate change and improve practices within communities and organizations [Lew46]. In the field of information systems, AR has evolved to address both technological and organizational challenges, making it a flexible and relevant approach for solving real-world problems while advancing scientific knowledge [Bas99]. The core aspects of AR include:

- **Practical Problem Solving and Theoretical Progress:** AR focuses on solving practical problems while simultaneously contributing to theoretical knowledge [BM04, POS12].
- **Iterative Cycles:** The cyclical nature of the method allows for continuous refinement and adaptation of approaches, fostering ongoing learning and improvement [Bas99].
- **Pragmatic Foundation:** Rooted in pragmatism, AR emphasizes creating actionable knowledge that has direct applicability [POS12].
- **Contextual Sensitivity:** By focusing on the specifics of the context, AR ensures that its findings are relevant and tailored to the environment being studied [CGM09].

- **Comprehensive Methodological Approach:** The method combines empirical investigation with conceptual development, adapting to the needs of the situation [POS12].
- **Learning and Knowledge Production:** AR emphasizes generating knowledge through action and reflection, leading to continuous learning and improvement [Lew46].
- **Active Intervention:** The approach involves researchers actively implementing changes and observing their effects in a real-world environment [Bas99].

3.2 Rationale for adopting the AR-Methodology

Based on the core aspects outlined above, we opted to apply AR in our work for several reasons:

AR's emphasis on **practical problem-solving and the advancements of theoretical knowledge** [BM04] aligns seamlessly with the objectives of our study. Since our work aims to evaluate the extent to which LLMs are currently able to perform BPO, its results are not only of interest to the academic audience but also to businesses seeking to enhance operational efficiency in a less resource-straining way than it has been possible previously. Thus we argue that our work bridges the gap between academia and practice, which aligns well with the principles of AR.

We further argue, that our work is also closely aligned with AR's aim of **creating positive change** due to its role as an enabler of easing process optimizations, thus driving increased efficiency and effectiveness. The **iterative nature** of AR, with its focus on reflective learning and knowledge production through iterative cycles, is particularly well-suited for working with LLMs. Given that interactions with LLMs often entail a conversational nature, leveraging an iterative methodology which allows for allowing for continuous refinement of the approach and thus improved results, seemed ideal. In contrast, methodologies that consider only a single cycle, such as design science [HMPR04], would have been less effective for our research. This is caused by the fact that when building our research upon them, we would not have been allowed continuously adapt and evolve our approach based on the outputs of the model, possibly leading to shallower insights and thus inferior results.

The methods focus on **contextual sensitivity** aligns well with the context-dependent nature of LLMs (see section 2.2). Considering that the performance of LLMs varies significantly based on the context in which one interacts with them, having this characteristic integrated into our method ensures that our research inherently considers this variability. This ensures that our findings are not only theoretically valid but also practically relevant and applicable.

One aspect of AR, where our approach, as it will be outlined below (see section 3.3), does on the first glance not meticulously follow the classical AR method is its approach towards

active intervention. With AR's origins in social sciences, classical interpretations of it consider the interaction with the studied people when talking about the interventionist approach [Bas99]. In our work, however, the interaction centers on the LLM. We argue that this approach still adheres closely to the methodology, as the LLM under study exhibits a similar influence to that of human subjects in comparable research. Through our interactive engagement with the LLM, we derive insights that guide and inform subsequent research cycles. This ensures that the intervention remains a dynamic and integral part of the AR process.

Our **review of the scientific literature** demonstrated that other authors have reached similar conclusions for related studies. Herdel et al. for example, followed a classical interpretation of AR in developing a framework for assessing AI use cases and associated risks in an interventionist manner by interacting with AI-practitioners [HSBQ24]. Differently, li et al. took an approach more similar to ours as they focused on the iterative nature of AR and interacted mainly with the LLM in order to derive a framework for data augmentation through LLMs [LZZ$^+$24].

3.3 Application of the Methodology in this work

In this section, we will elaborate on how each step of the AR methodology will be meticulously followed in our research. The graphic provided below (see Figure 3.1) offers a visual representation of our methodical approach.

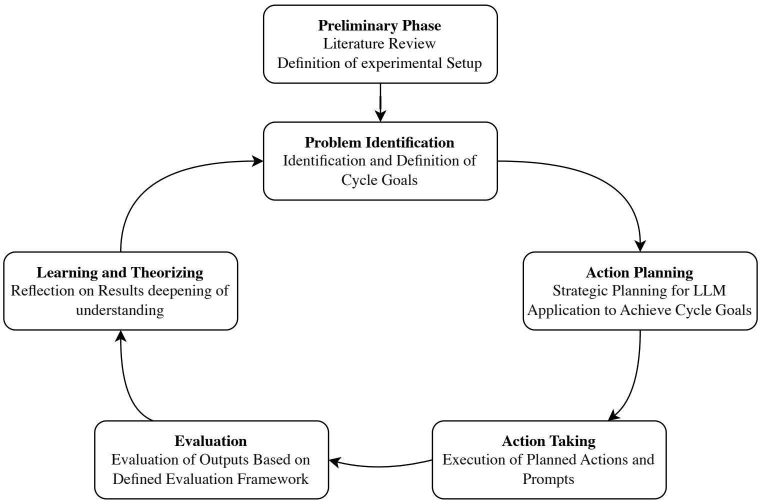
Preliminary Phase
Literature Review
Definition of experimental Setup
Problem Identification
Identification and Definition of Cycle Goals
Action Planning
Strategic Planning for LLM Application to Achieve Cycle Goals
Action Taking
Execution of Planned Actions and Prompts
Evaluation
Evaluation of Outputs Based on Defined Evaluation Framework
Learning and Theorizing
Reflection on Results deepening of understanding

Figure 3.1: Our Research Process

As can be seen in the graphic, the research process encompasses an initial Preliminary Phase followed by multiple iterative Action Cycles, with each of them building on the results of the previous one.

3.3.1 Preliminary Phase

This preemptive phase serves as the entry point of our research. It provides a grounding for our work and specifies the finer details of the Research Framework that will be applied in the action cycles.

A Literature review of extensive nature represents the first phase of our work. This review focuses on BPM and LLMs, as well as existing research on their intersection. We conducted a systematic literature search using databases such as *Springer* [Spr], *ArXiv* [ArX], and *ResearchGate* [Res]. Specific keywords and phrases used in our search included "*Business Process Management*", "*Large Language Models*", "*automation BPO*," "*process optimization LLM*," and "*Large Language Model business processes*". This approach ensured a targeted and comprehensive review, detailed in chapter 2, which identifies key trends and research gaps that form the foundation for the subsequent stages of our work.

The definition of the experimental setup marks the second part of this preliminary phase. In this stage, the framework for executing and evaluating the results of each cycle is detailed. First, requirements towards each component of the research are defined based on the insights gained through the literature review. Next, decisions are made, and the rationale behind each of them is provided. For details on this phase, please refer to chapter 4.

3.3.2 Action Cycles

Following the preliminary phase, the principal segment of our research will be conducted in accordance with the methodological steps delineated by Baskerville and Wood-Harper in [BWH98]. Each cycle will comprise five phases depicted in the Figure 3.1 and outlined below:

Problem Identification and Contextual Understanding is the first phase of each cycle and involves understanding and defining the problem within its context. In this work, this phase will take on the form of a comprehensive examination of our current insights on the capabilities of LLMs for BPO as well as the determination of the cycle's objectives. The initial cycle will be based on insights from the literature, while subsequent cycles will reassess and refine these understandings based on the outcomes of previous stages, adapting the focus and approach as necessary.

Action Planning is the phase in which actionable strategies are developed to address the identified problems. In our work, this involves planning and preparing the manner in which LLMs will be applied to the process model to achieve the desired results by building on the knowledge gained from the literature and from previous cycles.

Action Taking considers implementing the planned actions in the context of the identified problems. In our work, this takes on the form of prompting our task to the LLM, in the manner that has been strategized in the previous phase.

Evaluation is the penultimate step of each cycle and is of critical importance in AR as it provides feedback on the implemented actions. In our work, this step involves a systematic assessment of the process model adaptations provided the LLM as a results of the action taking phase, in the light of the developed evaluation framework (see section 4.3).

Learning and Theorizing is the final phase of each cycle. It involves reflecting on the outcomes of each cycle and theorizing based on its findings. This reflective analysis will help refine the understanding of LLM capabilities for the given BPO task and contribute to broader knowledge. The subsequent cycles will then, in alignment with the principles of AR, leverage these insights in order to guarantee continuous improvement and an achievement of deeper insights with each cycle.

CHAPTER 4

Requirements Assessment and Decision Rationale

This chapter outlines the requirements for applying LLMs to BPO tasks, derived from the literature and conclusions presented in chapter 2. For each requirement, we provide a rationale for how our approach addresses it. Key decisions, including the selection of the business process, process representation, LLM model, prompting strategy, as well as the resulting evaluation framework, are explained to ensure both practical relevance and scientific rigor.

4.1 Business Process

In this section, we delineate the requirements that a business process must satisfy to be suitable for inclusion in our study, while providing a rationale for the corresponding decisions. Initially, we justify the selection of a digitization project process as the focal domain of this research, followed by a discussion of our decision to provide the process to the LLM in the BPMN format.

4.1.1 Rationale for Process Selection

The Requirements that we pose to the Process have been determined based on what was presented in section 2.1 and ensure that the chosen process is well aligned with the objectives of the study. These requirements are:

Proc. Req. 1 **Practical Relevance**: The process should be relevant to current organizational challenges, ensuring the studies impact by demonstrating the capabilities of LLMs. Therefore, we require processes that are actively used in a real-world context, as opposed to purely academic examples, which may not accurately reflect current industry practices.

Proc. Req. 2 **Complexity**: The process must present a sufficiently comprehensive challenge, allowing the LLM to showcase its capabilities across tasks of varying complexity. Consequently, the process should involve multiple interdependencies and intricate elements to provide a robust evaluation framework.

Proc. Req. 3 **Availability**: Naturally, the process must be accessible for the purposes of this study, which imposes certain constraints on our selection and must be explicitly acknowledged.

A digitization project process was chosen based on the outlined requirements. Digitization is highly relevant to current organizational challenges, as businesses must adapt to increasingly competitive, efficiency-driven economies. By focusing on digitization projects, this study directly addresses real-world issues faced by many organizations, thereby fulfilling the demands of Proc. Req. 1 and ensuring that the research aligns with actual industry needs. Digitization project processes inherently address the complexity requirement outlined in Proc. Req. 2, as these projects encompass a range of activities, from planning and design to implementation and evaluation, they provide a robust framework to test different aspects of LLMs for BPM optimization. We were able to ensure the availability of such a process, fulfilling Proc. Req. 3, by collaborating within a regional network of scientific institutions that supports research and innovation through a shared collaborative IT ecosystem. Since this process is actively used in the network's day-to-day operations, we can confirm its practical relevance, demonstrating that it is grounded not only in academic theory but also rigorously tested against practical business needs.

The main process of this digitization project is depicted in Figure 4.1, demonstrating its fulfillment of Proc. Req. 2 through the use of several subprocesses and decision points.

4.1.2 Rationale for Process Representation

The Requirements to the Process Representation that we pose are once again aligned with the observations made in chapter 2 and influenced by the choices made for the process. These requirements are:

Repr. Req. 1 **Practical Relevance**: The most pressing requirement is that the representation must be used in practice. It is often the case that strict modeling standards are not followed in practice and are adapted to meet the needs of the organization in question.

Repr. Req. 2 **Representation of Complexity**: The representation must be able to represent the complexity that our chosen process entails.

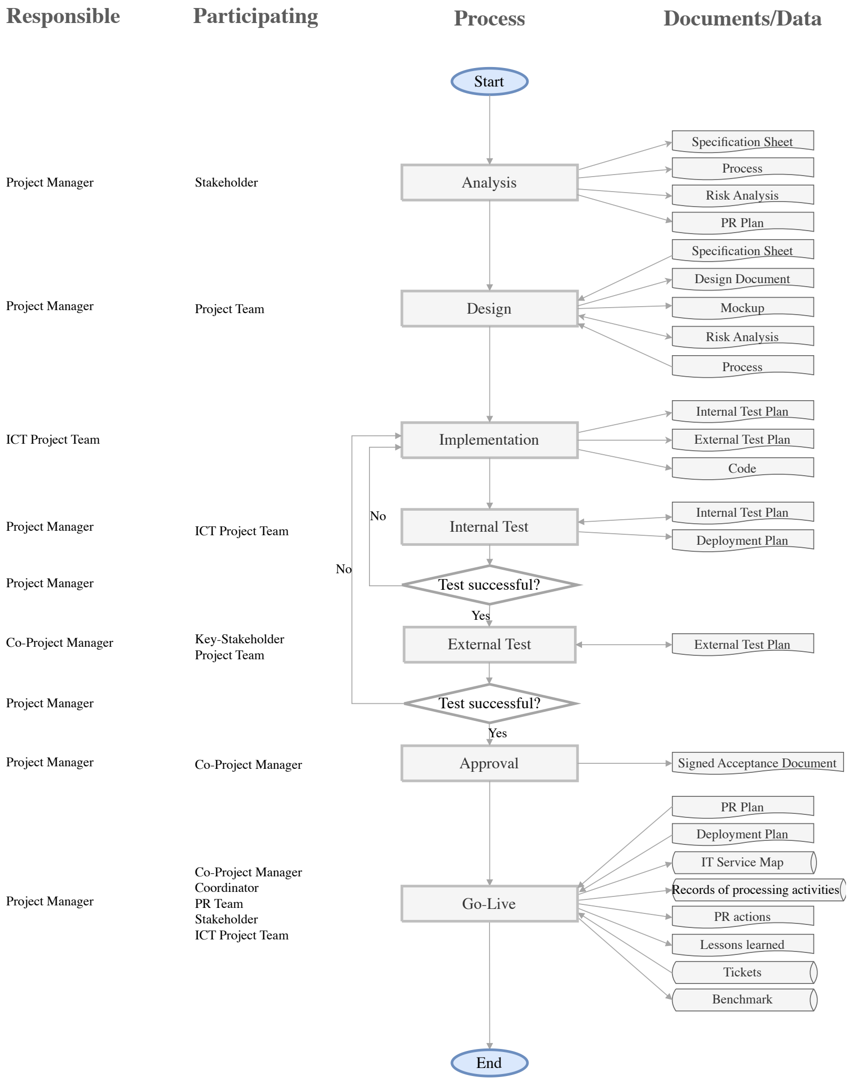

Figure 4.1: Main Process of the digitization process

Repr. Req. 3 **General Understandability**: Stakeholders within the organization and, in the context of our work, the scientific community, must be able to understand the representation.

A BPMN Model was selected based on the outlined requirements, primarily due to the practical relevance of BPMN models, which are widely adopted in industry and can be tailored to meet organizational needs, thus addressing the criteria in Repr. Req. 1. The process used in this study adopts an adapted form of BPMN. As shown in Figure 4.1, the process omits swimlanes, a deviation from the standard, based on feedback from process owners who observed that swimlanes often cause stakeholders to focus exclusively on their own sections, rather than viewing the process holistically. Moreover, BPMN's ability to represent processes with varying degrees of complexity was another key factor in our selection. Given the complexity requirement outlined in Repr. Req. 2, it was essential that our chosen process modeling approach supports detailed representations, allowing us to capture the interdependencies and nuances within the process. By contrast, representing processes in a table format, which is commonly used in companies, would not have sufficed for our study, as tables tend to oversimplify processes, often missing key elements like decision points and other complex aspects crucial for accurately reflecting the flow of process execution, thus providing only a high-level overview. Additionally, BPMN meets the requirement for understandability as specified in Repr. Req. 3. As a flexible and adaptive industry standard for business process modeling, BPMN is widely recognized and easily understood by stakeholders, even those with limited technical backgrounds, due to its clear and intuitive graphical format. The process used in this study demonstrates this adaptability, having been adjusted to prioritize understandability for employees over strict adherence to all BPMN standards. Nonetheless, it provides a realistic and highly relevant example of a complex, real-world process, effectively balancing usability with practical modeling requirements.

4.2 Large Language Model

In this section, we outline the requirements established for the LLM employed in our study, alongside the rationale behind our key decisions. Initially, we present the justification for selecting *GPT-4o* as the model for this research; subsequently, we elaborate on the reasoning behind utilizing basic prompting techniques.

4.2.1 Model Selection

The Requirements to the LLM that we pose are based on the aspects presented in chapter 2 and the decisions taken above for the Process (see section 4.1). The requirements delineated requirements are:

Model Req. 1 **Accessibility**: The model must be easily accessible to users. Since one of the cornerstones of our research is its applicability in real organizations, we need to focus on models that are easily accessible and, in the best case, already in use in businesses.

Model Req. 2 **Performance**: To ensure that our results are impactful and innovative, we need a SOTA model. This is essential to accurately assess the current capabilities of LLMs in the field, given the speed at which the field is evolving.

Model Req. 3 **Multimodality**: To ensure the model's usability meets our expectations, it must be capable of processing and interpreting various real-world data formats, extending beyond text to include images, among others. This requirement stems directly from our earlier decision (see subsection 4.1.2) to provide our process models to the LLM in BPMN format.

GPT-4o through ChatGPT [Ope24c] is the choice we have made based on the requirements presented. This decision is based on several factors. Through the *ChatGPT* platform, the model is easily accessible to all types of users, fulfilling the accessibility requirement (see Model Req. 1). Users can use the model it through a free membership on *ChatGPT* [Ope24a], which has a limited number of prompts per hour, or a paid *ChatGPT* membership for advanced features and no interaction limits. In addition, developers can access, customize, and integrate the model into their services through an API that includes a pay-per-token pricing. Furthermore, the model is integrated into *Microsoft365 Copilot*, which uses it in the backend to improve productivity in widely used applications integrated into *Microsoft365* [Mic23b, Mic24b]. Given the importance of *Microsoft365* in business environments, using a similar model aligns our work with tools that actual end users are likely to encounter [Mic23a]. Furthermore, this requirement also leads us to use a general-purpose model rather than a specialized one, or even to fine-tune a model ourselves. Given that it has been shown on medical question-answering benchmarks that general-purpose models can outperform fine-tuned specialized models under certain conditions, we argue that this decision does not automatically degrade the quality of our results [NLZ+23]. In terms of our performance requirements, *GPT-4o* provides the advanced capabilities necessary to deliver impactful and innovative results [Ope24c]. As one of the current SOTA models, alongside others such as Meta's *Llama 3.1* [AI24b], Alphabet's *Gemini 1.5* [AI24a] or Anthrophic's *Claude 3.5* [Cla24], *GPT-4o* excels in language understanding and generation, making it ideal for our research needs. This version of GPT also satisfies the multimodality requirement (see Model Req. 3). Unlike earlier models such as *GPT-3.5* [Ope24b], which focused primarily on text, *GPT-4o* is designed with a specialized architecture that allows it to process and understand multiple data formats, including text and images. In addition, as described in section 2.3, Lins et al. [LNA+23] have demonstrated that *GPT-3.5* can understand and provide execution support for XML process models. Given the enhanced multimodal capabilities of *GPT-4o*, we aim to move beyond XML files and use graphical input, as outlined above. In summary, although there are other notable models that could be used, due to the strong adaptation that GPT models have experienced in practice and in research (see section 2.4) and due to the fact that they meet all the requirements that we have posed, we have chosen to use them for our study instead of using another model.

4.2.2 Rationale for Prompting Strategy

Requirement to the Prompting Strategy

Req 1 **Practical Relevance**: The real-world focus of our work demands that the prompts that we use are similar to what users in real-world organizations would be using.

A simplistic prompting strategy was chosen based on the requirement for practical relevance outlined in Req 1. In real-world applications, as discussed in section 2.2, most users tend to use basic prompting techniques that often lead to suboptimal results [Cai24, S$^+$22]. Despite the availability of more sophisticated techniques such as multi-agent reasoning [LZY$^+$24] and chain-of-thought [JYS$^+$24], these methods are relatively unfamiliar to the broader user base. Since our research emphasizes the practical applicability of LLMs in BPO tasks, we deliberately avoid these advanced strategies. By focusing on simpler prompting methods, we want to ensure that our findings are relevant to typical end users who are unlikely to use more complex techniques in practice.

4.3 Evaluation

The Requirements to the Evaluation Drawing upon the studies reviewed in chapter 2 and analyzing their evaluation methodologies, we have derived the following requirements for assessing the capabilities of LLMs in our research:

Eval. Req. 1 **Evidence-Based Criteria**: The evaluation should be based on criteria that are supported by empirical evidence and academic research. This ensures that the best practices are not only theoretically sound but also proven in practice.

Eval. Req. 2 **Adaptability to Context**: The framework must be adaptable to the specific context of the organization and its processes. Such adaptability ensures that the evaluation remains relevant and customized to the unique characteristics of the processes, thereby enabling precise and context-specific conclusions.

Eval. Req. 3 **Feasibility**: The framework must be practical to implement, with criteria that are assessable within the organization's resource limitations. This ensures that the evaluation can be carried out both efficiently and effectively, without exceeding available resources.

Eval. Req. 4 **Scalability**: The framework must be scalable, enabling its application to processes of varying sizes and complexities, thus ensuring its applicability in future studies as well.

A best-practice-based evaluation framework has been selected for our study based on the criteria outlined above. In this framework, we systematically assess the capabilities of LLMs in performing various BPO-related tasks by analyzing the degree of deviation between the LLM-proposed solutions and the outcomes expected according to established best practices. This approach allows us to determine the extent to which

LLM-generated outputs align with these established standards, enabling scientifically valid and practically relevant conclusions, fulfilling the requirement of Eval. Req. 1 and responding to calls from the scientific literature for clear evaluation criteria that are reproducible when working with LLMs in research [ETdROR24]. The adaptability of this approach can be used to evaluate the LLM's ability to perform different BPO tasks using the same framework, meeting the requirements outlined in Eval. Req. 2. The approach is also feasible in terms of our study, considering that, as outlined in chapter 3, due to time and resource constraints, we are not able to actually implement process changes in real organizations and then measure quantitatively if the proposed processes result in higher efficiency. However, the best-practice evaluation approach is feasible in our work and thus meets the requirement of Eval. Req. 3. Finally, our approach is also scalable, as required by Eval. Req. 4. Considering that we are evaluating the ability of LLMs to perform different specific tasks that are a part of process optimization, our approach can be applied to processes of different size and complexity, allowing us to adapt our framework to the given process if necessary.

4.3.1 Resulting Evaluation Framework

Based on the considerations outlined above, as well as those discussed in chapter 2, we have developed the following evaluation framework, which integrates into our research process from chapter 3, as depicted in Figure 4.2. The framework primarily focuses on two key phases from the AR cycle outlined in section 3.3: the *Action Planning Phase* and the *Evaluation Phase*.

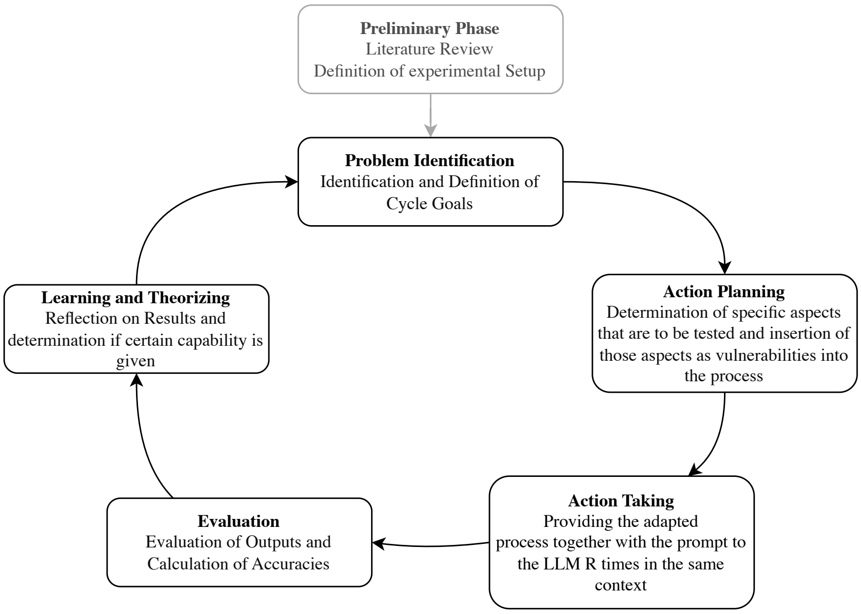
Preliminary Phase
Literature Review
Definition of experimental Setup
Problem Identification
Identification and Definition of Cycle Goals
Action Planning
Determination of specific aspects that are to be tested and insertion of those aspects as vulnerabilities into the process
Action Taking
Providing the adapted process together with the prompt to the LLM R times in the same context
Evaluation
Evaluation of Outputs and Calculation of Accuracies
Learning and Theorizing
Reflection on Results and determination if certain capability is given

Figure 4.2: Research Process detailed for the Evaluation Framework

In the **Action Planning Phase**, we identify specific aspects of LLM capability to evaluate, guided by the cycle's objectives. This involves selecting a total of $BP = 5$ best practices from the scientific literature, which are intentionally violated in I instances within the modified process. The value of I is chosen to ensure a balanced evaluation across all BP best practices, maximizing the comprehensiveness of the assessment. The altered process, together with the corresponding prompt, is then presented to the model during the *Action Taking Phase*. Although constructed examples are utilized, we deliberately apply a real-world digitization project process to introduce the complexity and variability typical of actual business processes. This approach ensures that the evaluation reflects real-world conditions, which purely constructed examples may fail to capture, thereby enhancing the practical relevance of our findings.

In the **Evaluation Phase**, we assess how many of the introduced I instances of optimization potentials from the BP best practices the LLM identifies across $R = 5$ runs of the same process model and prompt, all conducted within the same context. Multiple runs are required due to the inherent randomness configured into the outputs of most LLMs including the *ChatGPT* environment utilized in this study, through a so-called *temperature* parameter. This non-deterministic behavior necessitates repetition to obtain a robust understanding of the model's overall capabilities. To quantitatively evaluate the model's performance over a singular process, we compute an *Accuracy Score*, which reflects the normalized performance across R repetitions and I best practices in the specific process. The score is calculated as:

$$\text{Accuracy} = \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \mathit{Allocated\ Points}_{i,r} \tag{4.1}$$

The *Allocated Points* are awarded based on the model's performance in each of the BP tasks across the R runs, using the following criteria:

- *1 pt*: The intended aspect is clearly addressed in a detailed manner, utilizing the intended best practice.
- *0.5 pts*: The intended aspect is addressed but lacks specificity or is solved in a valid manner that however utilizes a different best practice than intended.
- *0 pts*: The intended aspect is not addressed in any meaningful way.

By analyzing the results of our experiments, we can provide a single, quantitative measure of the model's capabilities.

This framework, which integrates real-world processes, multiple evaluation iterations, and clear scoring metrics, ensures a comprehensive assessment of the LLM's capacity to perform BPO tasks. It evaluates the model's adherence to established best practices while accounting for the variability in its outputs, offering both practical relevance and scientific rigor. Consequently, this approach provides a robust understanding of the model's strengths and limitations in optimizing business processes.

CHAPTER 5

Adherence to Modeling Criteria

This chapter addresses SRQ1 through a two-cycle analysis of LLM capabilities in identifying process modeling issues. The first cycle uses a general prompt, while the second employs a context-enhanced approach with a prompt that explicitly references the *Seven Process Modeling Guidelines (7PMG)*. Our findings show that LLMs are moderately effective with general prompts but achieve high accuracy with explicit guidance, demonstrating their ability to adhere to business process modeling standards. These results positively answer SRQ1 and provide a basis to answer the remaining research questions in the subsequent chapters.

5.1 Assessment of Process Model Quality

5.1.1 Problem Identification and Contextual Understanding

As outlined in chapter 1 and chapter 2, current automation tools for BPO fall short in meeting the evolving demands of modern business environments, often lacking the necessary speed and adaptability to deliver timely and effective solutions. Given the demonstrated potential of LLMs across various domains, including their capabilities in several phases of the BPM lifecycle, it is a logical progression to investigate their applicability in enhancing BPO. However, as highlighted in chapter 2, a significant gap remains in the scientific evaluation of LLMs's capabilities within the BPO context, making a systematic examination of this potential application imperative.

Building on these research gaps, our initial step is to determine whether LLMs are not only able to comprehend and process business processes, as demonstrated by Lins et al. [LNA^{+}23], but also to adhere to established modeling criteria. Specifically, we seek to ensure that process models generated by LLMs are constructed according to established principles, thereby testing the syntactic capabilities of LLMs in the context of BPO. Therefore, in this cycle, our focus is on evaluating the ability of LLMs to identify issues in process models and propose solutions to rectify these flaws. While the ability to identify and address such issues does not guarantee that LLMs will always conform to these standards, it does suggest a foundational understanding of the principles that govern sound process modeling. By examining their capacity to detect and articulate specific modeling deficiencies, we aim to assess whether LLMs are aware of and able to apply established modeling criteria effectively.

5.1.2 Action Planning

The best-practice standard and its guidelines that we will leverage in our evaluation, as outlined in chapter 4, must first be selected from the various standards available in the literature that focus on the correct modeling of business processes. For this study, we have chosen Mendling et al.'s *7PMG* [7PM10]. This choice is primarily motivated by the fact that these guidelines form the foundation for much of the literature used to teach sound process modeling, suggesting that they are likely to be at least partially adhered to in real-world applications, thereby ensuring a degree of practical relevance.
The *7PMG* emphasize reducing complexity, maintaining clarity, and ensuring consistent notation, which are crucial for effective stakeholder communication and process analysis. From the presented guidelines, as outlined in subsection 4.3.1, we will use the following $BP = 5$ representative examples as best practices for our study:

BP1: *Use as few model elements as possible*
Focuses on minimizing the number of elements in a model to reduce complexity, improve clarity and reduce cognitive load, as larger models are more difficult to comprehend and prone to higher error rates [MRC07, MVvD^{+}08].

BP2: *Minimize the routing paths per element*
Recommends limiting the number of input and output arcs associated with each element as complex routing increase the likelihood of errors and decreases model comprehensibility [MRC07, MNvdA07].

BP3: *Use one start and one end event*
Ensures that the process model has a clear entry and exit point, reducing ambiguity and making the model more accessible for process analysis tools [MNvdA07, vdAtHKB03].

BP4: *Model as structured as possible*
Advocates for structured modeling where every split has a corresponding join, creating a balanced flow, increasing accuracy and reducing the risk of misinterpretation [MRC07, MNvdA07].

BP5: *Use verb-object activity labels*
Encourages the use of consistent, verb-object activity labels (e.g., `Submit Report`), reducing ambiguity and making the process easier to interpret [MRR, RM08].

The process that will be leveraged for this experiment is the *Analysis* sub-process of the digitization project process that was provided to provided to us, as detailed in subsection 4.1.1. We have modified the process by introducing $I = 5$ violations of the best practices specified above. In this case, it follows that $BP = I$, as we are using only a single process for this analysis. Specifically, we have added excessive and overly detailed steps, introduced unnecessary gateways, routing paths and events, and altered labels to make them overly detailed and lengthy. A comparison between the original *Analysis* sub-process (see Figure 9.1) and our modified version (see Figure 9.2) is shown in subsection 9.1.1.
We provide the process to the LLM in PDF format. This decision is grounded in the central role that the format has in real-world business environments, where business process models are frequently shared, analyzed, and stored in PDF format for easy accessibility and standardized presentation.

The prompt provided to the LLM for analyzing the adapted process is the next aspect that has to be addressed. Given our decision to use simple prompts (see subsection 4.2.2), we will not employ advanced tactics such as agent-based approaches or few-shot learning. Thus, we have chosen the following prompt:

> Analyze the correctness of the modeling of the attached process model, focusing on identifying elements that do not adhere to best-practice modeling standards. Highlight any inconsistencies, redundant gateways, unclear connections, misused symbols, excessive complexity, or any deviations from standard modeling conventions that could be improved for clarity, simplicity, and accuracy.

Prompt 1: Assessment of Process model quality

This prompt is deliberately minimalistic, consistent with the decisions made in subsection 4.2.2 to assess the general capabilities of LLMs rather than those guided by expert-driven prompts. It is also important to note that we specifically avoid mentioning the *7PMG* in this prompt to prevent biasing the LLM towards these guidelines, allowing us to observe whether it considers them organically.

5.1.3 Action Taking

The planned strategy is then executed by prompting the LLM as specified and providing the adapted process to it in PDF format. This procedure was repeated, as defined in subsection 4.3.1, $R = 5$ times to mitigate the effects of model randomness and ensure robustness in the results. The outputs generated in response to each prompt are presented in subsection 9.1.2, specifically in LLM-Output 1, LLM-Output 3, LLM-Output 5, LLM-Output 7, and LLM-Output 9.

5.1.4 Evaluation

Below, a detailed analysis of the model's responses is presented, following the approach outlined in subsection 4.3.1. In the specific case of this analysis, the defined points allocation can be further specified as follows way:

- *1 pt*: The violation of the *LLM* is clearly identified and specific examples or detailed feedback that directly addresses the guideline and it's violation are provided.
- *0.5 pts*: The violation of the *LLM* is mentioned, however lacking specificity or depth.
- *0 pts*: The violation of the *LLM* is not mentioned at all.

Based on the review of the LLM outputs provided in section 9.1, the model responds to the violations of the *7PMG* in the following ways:

- *Use as Few Elements as Possible (BP1):* In two instances, the model clearly identifies redundant gateways and tasks, providing specific suggestions to eliminate them. In the remaining three instances, the model mentions redundancy in decision points and roles but lacks specificity, offering general observations without detailed examples of where improvements can be made.
- *Minimize the Routing Paths per Element (BP2):* In three instances, the model recognizes overly complicated paths and suggests simplification by reducing the number of routing paths. In the other two cases, while the model mentions path complexity and overlapping paths, it provides more general feedback, without detailed suggestions for improvement.
- *Use One Start and One End Event (BP3):* The model clearly identifies multiple start and end events in one instance, recommending consolidation to a single start and end point. In the other four instances, it notes inconsistencies in the use of start and end events but provides less detailed or clear analysis of the issues.
- *Model as Structured as Possible (BP4):* In two instances, the model provides comprehensive analysis of structural problems, highlighting areas where unstructured decision points and routing create confusion. In the remaining three instances, the model mentions the need for restructuring but does not provide specific examples or offer clear guidance on improving the process structure.
- *Use Verb-Object Activity Labels (BP5):* In two instances, the model identifies unclear activity labels and suggests the use of verb-object formatting to improve clarity. In the other three instances, the model either fails to mention the issue or provides general comments on label ambiguity without specifically addressing the need for verb-object labels.

Best Practice	LLM-Output 1	LLM-Output 3	LLM-Output 5	LLM-Output 7	LLM-Output 9	Average
BP1	0.5	0.5	0.5	1	1	**0.7**
BP2	1	1	1	0.5	0.5	**0.8**
BP3	0.5	0.5	1	0.5	0.5	**0.6**
BP4	1	0.5	0.5	1	0.5	**0.7**
BP5	1	0.5	0	1	0.5	**0.6**

Table 5.1: Points Allocation for Recognition of 7PMG Violations

Based on these insights, we have awarded the following points:

With the number of rows in the table equal to $I = BP$ and the number of columns of LLM-Outputs equal to R.

This yields the following value for Equation 4.3.1, representing the model's overall accuracy in recognizing *LLM* violations in this experiment, where *Allocated Points* refer to the points specified in the table above:

$$\text{Accuracy}_{\text{7PMG}} = \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points}_{i,r} = \frac{1}{5 \times 5} \times 17 = 0.68 \qquad (5.1)$$

Thus, the analysis yields an overall accuracy score of 0.68, reflecting the model's varying ability to recognize violations of the chosen best practices.

The evaluation reveals that the model performs best in recognizing flow-related issues (*BP2*), where it consistently identifies complex routing paths and suggests improvements over all repetitions, achieving an average score of 0.8.

The model's performance in structural optimization is solid but fluctuates, with scores for (*BP1*) and (*BP4*) averaging 0.7. Although the model occasionally provides detailed feedback, its responses vary across iterations, leading to general or unspecific recommendations at times.

In contrast, the model struggles to deliver in-depth analysis on more subtle issues, such as start/end events (*BP3*) and labeling (*BP5*). It often provides superficial feedback or omits key details, resulting in an average score of 0.6 for both guidelines.

5.1.5 Learning and Theorizing

The evaluation shows that while the model reliably identifies element-wise violations such as *BP2*, its performance fluctuates across iterations, particularly when addressing more complex aspects like *BP1* and *BP4*, which involve analyzing relationships between multiple elements.

Additionally, the recurring issue of superficial feedback and the model's tendency to generalize problems is especially evident in guidelines like *BP3* and *BP5*, where the model often identifies issues but delivers vague or generalized recommendations.

Moreover, there is a noticeable bias toward common violations, with the model performing well at detecting explicit issues like routing problems (*BP2*), but struggling to recognize more subtle violations, such as unclear labeling in *BP5*. This indicates that the model is better suited for identifying obvious errors rather than addressing more intricate modeling challenges.

In light of these insights, and guided by previous studies results (see chapter 2), we theorize that providing the model with explicit contextual guidance, such as mentioning the *7PMG* in the prompt, could substantially enhance its performance. With clear references to these guidelines, we expect the model to more accurately identify violations and deliver more precise, contextually relevant solutions.

5.2 Context-Enhanced Assessment of Process Model Quality

5.2.1 Problem Identification and Contextual Understanding

As revealed in the first cycle, the model's performance was limited by its inconsistency, superficial feedback, and a tendency to overlook more nuanced violations, such as labeling issues. To address these gaps, we will explicitly incorporate the *7PMG* into the prompts for this cycle. By providing clearer guidance through direct references to these guidelines, we aim to enhance the model's accuracy in identifying both common and more subtle violations, while also improving the consistency of its feedback across iterations.

5.2.2 Action Planning

The best-practice standard and it's guidelines as well as the process used in this second cycle will remain the same as in the first cycle, building on the previous findings, ensuring comparability with the previous results. The processes can once again be seen in subsection 9.1.1.

The prompt for this cycle is aimed at providing more direct and explicit guidance to the model. Unlike the previous cycle, where the LLM was expected to identify issues without specific reference to modeling standards, this time we explicitly incorporate the *7PMG* into the prompt, as outlined above.The prompt is formulated as follows:

> Focusing on the 7PMG, evaluate the provided process model by rating each guideline on a scale of 1-10, where 1 indicates poor adherence and 10 indicates excellent adherence.

Prompt 2: Assessment of Process model quality with Enhanced Context

This adjusted approach is intended to guide the model toward a more structured analysis, specifically targeting the *LLM*, to overcome the issues of the previous cycle, which are outlined above, and thereby enhance its overall *Accuracy Score.*

5.2.3 Action Taking

Using the adapted prompt but maintaining the same configuration as the previous cycle to ensure comparability, we conducted the experiments again, repeating the process $R = 5$ times. The outputs from these iterations are detailed in section 9.1, specifically in LLM-Output 2, LLM-Output 4, LLM-Output 6, LLM-Output 8, and LLM-Output 10.

5.2.4 Evaluation

Below, we once again give a detailed analysis of the model's responses, where we analyze the outputs, award points based on the same schema as in the previous cycle and calculate the new accuracy score.

Based on the review of the updated LLM outputs in section 9.1, the model's handling of the identified violations of the *7PMG* can be summarized as:

- *Use as Few Elements as Possible (BP1):* The model performs well in most instances, recognizing unnecessary elements and suggesting improvements in four out of five cases. In one instance however, it acknowledges the complexity but does not strongly criticize the excessive elements.
- *Minimize the Routing Paths per Element (BP2):* In four out of five cases, the model identifies overly complex routing paths and provides suitable recommendations for simplifying the process. However, one instance shows less specificity, where the model notes confusion but does not provide clear suggestions.
- *Use One Start and One End Event (BP3):* Across all instances, the model consistently identifies multiple start or end events and offers clear recommendations to simplify the model by using a single start and end event.
- *Model as Structured as Possible (BP4):* The model consistently identifies structural issues in the process model and provides valid recommendations for creating a more balanced and structured flow in all instances.
- *Use Verb-Object Activity Labels (BP5):* In three out of five instances, the model identifies and suggests improvements for inconsistent labeling. In one instance, it provides only partial recognition of the problem, while in the remaining one, it fails to address this issue completely.

As a result of this analysis, points are allocated in the following way:

With the same table format as above. We can now once again calculate the overall Accuracy Score for this cycle, using the same values for R and I as well as the updated *Allocated Points* from the table:

Best Practice	LLM-Output 2	LLM-Output 4	LLM-Output 6	LLM-Output 8	LLM-Output 10	Average
BP1	1	1	0.5	1	1	**0.9**
BP2	1	1	1	1	0.5	**0.9**
BP3	1	1	1	1	1	**1.0**
BP4	1	1	1	1	1	**1.0**
BP5	0	1	1	0.5	1	**0.7**

Table 5.2: Points Allocation for Context-Enhanced Recognition of 7PMG Violations

$$\begin{aligned}\text{Accuracy}_{\text{7PMGContextEnhanced}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points}_{i,r} \\ &= \frac{1}{5 \times 5} \times 22.5 = 0.9\end{aligned} \tag{5.2}$$

In this cycle, the results show a marked improvement compared to the first cycle, as the model recognized and addresses violations more consistently, resulting in a increased overall *Accuracy* of 0.9 compared to the previous cycle's 0.68.

The model now performs best in identifying multiple start and end events (*BP3*) and structural issues (*BP4*), achieving perfect scores of 1.0 across all iterations, a significant improvement over the previous cycle's averages of 0.6 and 0.7, respectively.

The model's performance in flow-related issues (*BP2*) and minimizing elements (*BP1*) has increased as well, with both averaging 0.9, an improvement over the first cycle's scores of 0.8 and 0.7, respectively.

Continued struggles are however encountered by the model in addressing labeling issues (*BP5*). The score increases from the previous cycle's 0.6 to 0.7, but the feedback remains somewhat superficial, often lacking depth in its comments when addressing the issue.

5.2.5 Learning and Theorizing

This cycle confirms that explicit guidance significantly enhances the model's ability to detect process modeling violations, particularly for clear, pattern-driven guidelines. Improvements in flow-related issues (*BP2*) and structural optimization (*BP4*) demonstrate how added context enhances recognition of explicit violations, addressing the inconsistency observed in the first cycle.

Despite these gains, however, the model continues to struggle with more abstract guidelines like labeling (*BP5*), similar to the first cycle. While context enhances the identification of pattern-driven violations, the model remains less effective in handling less systematic tasks like labeling.

The tendency to generalize complex relationships, particularly in minimizing elements (*BP1*), remains persistent when compared to the first cycle. Although the bias toward common violations has decreased, the model still performs best with easily recognizable patterns like routing issues (*BP2*), while continuing to underperform with nuanced violations such as labeling (*BP5*).

Overall, these findings confirm that LLMs can effectively generate content aligned with common process modeling criteria, thereby meeting the syntactic requirements of BPO to a satisfactory degree and providing a positive answer to SRQ1, which is detailed in section 7.1. This allows us to now shift the focus of the study in the subsequent chapter to evaluating the semantic capabilities of LLMs in executing BPO.

CHAPTER 6

Process Optimization Capabilities

This chapter addresses SRQ2 and SRQ3 through three additional cycles of AR, where we examine the semantic capabilities of LLMs in executing BPO, building on the insights gained from the previous two cycles, which focused on the model's syntactic capabilities, as discussed in chapter 5. The first cycle reveals limitations related to the PDF input format. In the second cycle, we address the limitations identified in the first cycle by using a bitmap input format, allowing us to overcome the format-related issues of the PDF input. This enables a more thorough evaluation of the LLM's performance on more complex tasks. Finally, the third cycle improves on the results of the second one, by incorporating organizational context into the prompt. This enhances the LLM's decision-making but introduces new challenges. These findings underscore both the strengths and limitations of the LLM in applying redesign heuristics, thereby addressing the research questions.

6.1 Assessment of Process Optimization Capabilities

6.1.1 Problem Identification and Contextual Understanding

Building on the findings from chapter 5, where it was demonstrated that LLMs can adhere to established business process modeling criteria from the scientific literature, this cycle examines their optimization capabilities in a focused manner. Adopting a divide-and-conquer approach, we will systematically assess the ability of LLMs to recognize and address process inefficiencies across individual optimization categories. This targeted analysis aims to evaluate the strengths, limitations, and overall feasibility of LLMs as tools for autonomous business process optimization, from a semantic perspective, focusing on optimization categories independently from each other, in order to avoid any correlations.

6.1.2 Action Planning

The best-practice standard and its guidelines used in this analysis are the *Redesign Heuristics* by Reijers et al. [Red05]. These heuristics were chosen for their broad recognition and dual applicability in both business and academia, providing a blend of academic rigor and practical relevance. Unlike other frameworks that may be overly theoretical or limited in scope, the *Redesign Heuristics* offer a balanced approach, addressing a wide range of optimization scenarios. This makes them, in our eyes, the most comprehensive and adaptable framework for evaluating the ability of LLMs to identify optimization potential and leverage them through process adaptations and thus they will serve as the benchmark to evaluate the LLM capabilities to recognize and suggest improvements that align with established best practices in process optimization.
Similarly to the last chapter, and following the approach outlined in subsection 4.3.1, we have once again chosen a subset of $BP = 5$ best practices, based on their impact and frequency in real-world processes:

BP1: *Task Elimination*
Focuses on the elimination of unnecessary tasks, particularly those that add no value from a customer's point of view, such as redundant control tasks or iterations. This heuristic has been widely cited in literature [PR95, BP97, VdAVH02].

BP2: *Task Composition*
Involves combining multiple smaller tasks into a single, composite task to reduce handoffs, setup times, and simplify processes. This heuristic is frequently mentioned as a method to optimize business processes [HC93, Buz96, SS97].

BP3: *Resequencing*
Refers to rearranging the order of activities to optimize the flow, minimize delays, and improve overall process performance [Kle95, VdAVH02].

BP4: *Parallelism*
Aims at performing independent activities simultaneously to reduce process cycle time and enhance throughput [RR94, Buz96, VdAVH02].

BP5: *Task Automation*
Refers to the use of technology to automate repetitive, rule-based tasks, reducing human intervention and increasing process throughput [PR95, HC93, BP97].

The processes which we will be using for this analysis can be found in subsection 9.2.1. Hereby, we adapted the process models which we were provided with by introducing elements to the process that present optimization opportunities according to the chosen

Best Practice	**Occurrences split through processes**			**Total Occurrences**
	Design	Implementation	Go-Live	
BP1	✓		✓	2
BP2		✓	✓	2
BP3	✓		✓	2
BP4	✓		✓	2
BP5		✓	✓	2
Total adaptations per process (I)	3	2	5	10

Table 6.1: Distribution of Optimization Opportunities Across Business Processes

Redesign Heuristics. An overview over the distribution of these alterations through the adapted processes can be seen in Table 6.1.

As shown in the table above, we have incorporated varying degrees of alterations and optimization potentials into the processes. This approach not only helps us assess how well the model applies specific *redesign heuristics* across different processes but also provides insight into how the complexity of a process influences the model's performance. It is expected that the model will find it easier to optimize processes with fewer intertwined optimization potentials compared to those with more.

We will once again provide the process to the LLM using to common PDF format, aligning closely with real-world practices, where PDF is also one of the most commonly used format in such an environment. Naturally, we will not provide the model with the marked optimization opportunities, as they are provided in subsection 9.2.1, but rather without them.

The prompt used in this cycle to assess the LLM's ability to optimize processes is once again of minimalistic nature, aligning with the strategy outlined in subsection 4.2.2. It is defined as:

> Optimize this process, stating what you changed

Prompt 3: Optimization with Minimal Context

By prompting using the word `optimize` we aim to push the LLM towards performing the whole BPO workflow, incorporating BPM's *analysis* and *redesign* phases, as outlined in section 2.1. Thus in the output we expect the LLM to provide optimization opportunities as well as how it would solve them in order to make the enhance the process. Notably, as in section 5.1, we do not mention our evaluation criteria in this first cycle. This approach aims to align the model's outputs as closely as possible with likely real-world results, without overly steering the model in any particular direction.

6.1.3 Action Taking

We then executed the planned strategy using the adapted processes by prompting the LLM as specified above. Similar to our approach in chapter 5, we repeated this approach $R = 5$ times to mitigate the effects of model randomness and ensure robust results.

During the execution of these tests however we noticed an issue in our approach, which has not shown itself clearly before and which lead us to abort this cycle after having only analyzed the *Design* Process. The results of this analysis can be found in section 9.2.2 in LLM-Output 11, LLM-Output 12, LLM-Output 13, LLM-Output 14 and LLM-Output 15.

6.1.4 Evaluation

In our tests, we noticed after a few iterations, that the model through all iterations had issues with understanding the process model in detail. It notably understood the general structure and elements of the representation, but through all outputs a lack of real understanding of the model, especially in terms of step order was recognized, making the evaluation of further processes superfluous and necessitating an adapted approach.

Notably, the model was still partially able to identify optimization opportunities within the processes, attempting to optimize them by resequencing or parallelizing steps, thereby applying several of the heuristics outlined above.

6.1.5 Learning and Theorizing

Based on the insights gained, we have observed that *GPT-4o* is currently incapable of optimizing processes when provided in PDF format. This limitation arises from the model's inability to correctly interpret the process structure within PDFs, likely due to the use of Optical Character Recognition (OCR) to convert the PDF's content into textual input. This conversion may result in misinterpretation of elements, potentially causing reordering and hindering accurate analysis and successful optimization.

While this initial attempt at process optimization did not yield the desired outcomes, it highlighted that the model possesses knowledge of, and attempts to apply, scientifically validated techniques, such as the *redesign heuristics*. However, due to the model's inability to accurately comprehend the process provided in the current format, a comprehensive assessment of its optimization potential could not be conducted.

Given these insights, and considering the advanced multimodal capabilities of current SOTA models, we theorize that if the processes were provided in a format better aligned with the model's input processing, bypassing OCR and preserving structural clarity, significantly improved results could be achieved, potentially demonstrating the full capabilities of LLMs for process optimization.

6.2 Assessment of Process Optimization Capabilities with adapted input method

6.2.1 Problem Identification and Contextual Understanding

To address issues related to the input formats identified in the previous cycle, we will transition the process model input from PDF to bitmap format [HJSM24]. This shift aims to eliminate OCR-induced misinterpretations, caused by the reliance on text extraction technologies for interpreting mixed-content PDFs. We thus aim to allow the LLM to directly utilize its multimodal neural network capabilities, enabling the LLM to preserve the structural integrity of the process model when applying the selected *redesign heuristics*, presumably yielding to improved results.

6.2.2 Action Planning

Considering the fact, that this cycle represents an adjustment from the previous cycle, we will not change any inputs apart from the format in which we provide the Process to the LLM. In order to do so, we export the BPMN process models into a PNG instead of a PDF format. The remaining choices regarding the best-practice standard, processes and prompt remain unchanged.

6.2.3 Action Taking

As before, we conducted $R = 5$ iterations of our experiments to mitigate the effects of model randomness, this time applying the process to all four adapted process models. The outputs generated in response to each prompt are presented in section 9.2.2.

6.2.4 Evaluation

Below, a detailed analysis of the models responses to each process and the introduced optimization potentials, once again following the approach outlined in subsection 4.3.1, can be found. For this analysis, the allocation of points has been further specified as follows:

- *1 pt*: The optimization potential is identified/solved using the intended redesign heuristic.
- *0.5 pts*: The optimization potential is identified/solved in a valid manner, but not using the intended heuristic or not on to the intended tasks.
- *0 pts*: The optimization potential is not identified/solved.

Design Process

- *BP1:* All outputs but LLM-Output 22 specifically state the introduced optimization potential and how it could be resolved using *task elimination.* The one outlying output does not state the issue, but the presented optimized process flow does not entail the redundant step anymore, thus still resulting in full points.

- *BP3:* All outputs correctly identify the optimization potential that was introduced into the process, but instead of focusing on eliminating repeated translation efforts by moving translation to after the mockup is fixated in `freeze mockup`, the model proposes to parallelize the translations with the other design development, focusing on maximizing throughput, which is a valid approach, but not the one that the original process follows.
- *BP4:* Over all iterations, only LLM-Output 16 and LLM-Output 24 recognize the parallelization potential of the risk analysis. However these steps also did parallelize the `risk analysis` with the remaining mockup development, but rather parallelize it with steps at the end of the process, like `freeze mockup`. This approach is also valid, but not the one that the original process followed.

Best Practice	**LLM-Output 16**		**LLM-Output 18**		**LLM-Output 20**		**LLM-Output 22**		**LLM-Output 24**	
	Found	Solved	Found	Solved	Found	Solved	Found	Solved	Found	Solved
BP1	1	1	1	1	1	1	1	1	1	1
BP3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
BP4	1	0.5	0	0	0	0	0	0	1	0.5

Table 6.2: Points Allocation for Found and Solved Opportunities: Design Process

The point allocations from the table provide the basis for calculating the Accuracy Scores. The Accuracy Score for the found optimization potentials, based on R repetitions and $I = 3$ optimization potentials that were introduced into this process, is determined using the *Allocated Points* in the *Found* column, resulting in:

$$\begin{aligned} \text{Accuracy}_{\text{Found}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Found)}_{i,r} \\ &= \frac{1}{5 \times 3} \times 9.5 = 0.633 \end{aligned} \tag{6.1}$$

Similarly, the Accuracy Score for the solved optimization potentials is determined using the *Allocated Points* in the *Solved* column, yielding:

$$\begin{aligned} \text{Accuracy}_{\text{Solved}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Solved)}_{i,r} \\ &= \frac{1}{5 \times 3} \times 8.5 = 0.566 \end{aligned} \tag{6.2}$$

Implementation Process

- *BP2:* The model successfully identifies and addresses optimization opportunities in two instances. However, in the remaining three cases, it proposes parallelizing the creation of the test checklist with the documentation or the implementation. While this suggestion is not inherently wrong, as the process model does not explicitly require the test checklist to be created before the documentation or after the implementation, it is not the path that the process follows, as it focuses on minimization of resource utilization instead of throughput maximization.
- *BP5:* The model recognizes the automation potential of this step in every repetition. While this step was quite obvious, it still marks the model's focus on automation of manual tasks.

Best Practice	**LLM-Output 26**		**LLM-Output 28**		**LLM-Output 30**		**LLM-Output 32**		**LLM-Output 34**	
	Found	Solved	Found	Solved	Found	Solved	Found	Solved	Found	Solved
BP2	1	1	0.5	0.5	1	1	0.5	0.5	0.5	0.5
BP5	1	1	1	1	1	1	1	1	1	1

Table 6.3: Points Allocation for Found and Solved Opportunities: Implementation Process

The point allocations provide the basis for calculating the Accuracy Scores in this case as well. The Accuracy Score for the found optimization potentials, based on R repetitions and $I = 2$ optimization potentials, is calculated using the *Allocated Points* in the *Found* column, resulting in:

$$\begin{aligned}\text{Accuracy}_{\text{Found}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Found)}_{i,r} \\ &= \frac{1}{5 \times 2} \times 8.5 = 0.85\end{aligned} \tag{6.3}$$

Similarly, the Accuracy Score for the solved optimization potentials is calculated using the *Allocated Points* in the *Solved* column, yielding:

$$\begin{aligned}\text{Accuracy}_{\text{Solved}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Solved)}_{i,r} \\ &= \frac{1}{5 \times 2} \times 8.5 = 0.85\end{aligned} \tag{6.4}$$

Go-Live Process

- *BP1:* The model failed to recognize the task `Internal approval of successful deployment` as redundant and not providing stakeholder value in any context.

Consequently, instead of eliminating this task, the model incorrectly merged it with other tasks, which not only retained the inefficiency but also prevented the correct tasks from being merged as intended, further hindering process optimization.

- *BP2:* The model's failure to identify the redundant task, `Internal approval of successful deployment`, significantly impeded its ability to effectively apply the *task composition* heuristic. As a result, instead of optimally merging the tasks `Deployment on Production System` and `Final test`, the model combined the redundant task with either of these, thereby failing to achieve the desired process optimization, but applying the correct *redesign heuristic.*
- *BP3:* in the instance, the model recognition that performing the `user satisfaction on analysis` before the `go-live` task was not feasible due to lacking user feedback that we will award zero points for both categories for every instance.
- *BP4:* The model identified the potential to parallelize the `PR Campaign` with other tasks in three instances. However, the original parallelization with the optional `Project Presentation` was applied in only two instances, while one instance incorrectly parallelized it with the `Final Test`. In another instance, where parallelization was not recognized, task combination was used instead, correctly combining tasks in a valid alternative approach. Therefore, 0.5 points are awarded for identifying and implementing task combination. The final instance made the `PR Campaign` conditional without further optimization.
- *BP5:* In every instance, the model accurately identified opportunities for applying *task automation* and correctly proposed optimizing the process through it.

Best Practice	**LLM-Output 36**		**LLM-Output 38**		**LLM-Output 40**		**LLM-Output 42**		**LLM-Output 44**	
	Found	Solved	Found	Solved	Found	Solved	Found	Solved	Found	Solved
BP1	0	0	0	0	0	0	0	0	0	0
BP2	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5
BP3	0	0	0	0	0	0	0	0	0	0
BP4	0.5	0.5	1	1	0	0	1	0.5	1	1
BP5	1	1	1	1	1	1	1	1	1	1

Table 6.4: Points Allocation for Found and Solved Opportunities: Go-Live Process

Based on the observations and allocated points, the Accuracy Scores for the optimization efforts on this process are calculated as follows. The Accuracy Score for the found optimization potentials, considering R repetitions and $I = 5$ optimization potentials, is computed using the *Allocated Points* in the *Found* column, yielding:

$$\begin{aligned} \text{Accuracy}_{\text{Found}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Found)}_{i,r} \\ &= \frac{1}{5 \times 5} \times 13.5 = 0.54 \end{aligned} \tag{6.5}$$

Similarly, the Accuracy Score for the solved optimization potentials is calculated using the *Allocated Points* in the *Solved* column, resulting in:

$$\begin{aligned}\text{Accuracy}_{\text{Solved}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Solved)}_{i,r} \\ &= \frac{1}{5 \times 5} \times 10.5 = 0.42\end{aligned} \tag{6.6}$$

Complete Analysis

Based on the scores defined above, the following Accuracy Scores are calculated for all three processes and the optimization potentials they entail. The Accuracy Score for the found optimization potentials, using R repetitions and the $I = 10$ optimization potentials that were introduced across all three processes, is determined by the *Allocated Points* in the *Found* column, resulting in:

$$\begin{aligned}\text{Accuracy}_{\text{Found}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Found)}_{i,r} \\ &= \frac{1}{5 \times 10} \times 31.5 = 0.63\end{aligned} \tag{6.7}$$

Similarly, the Accuracy Score for the solved optimization potentials is calculated using the *Allocated Points* in the *Solved* column, yielding:

$$\begin{aligned}\text{Accuracy}_{\text{Solved}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Solved)}_{i,r} \\ &= \frac{1}{5 \times 10} \times 27.5 = 0.55\end{aligned} \tag{6.8}$$

The insights from this cycle's analysis are summarized in the following table:

Best Practice	**Design Process**		**Implementation Process**		**Go-Live Process**		**Average**	
	Found	Solved	Found	Solved	Found	Solved	Found	Solved
BP1: Task Elimination	1	1	-	-	0	0	**0.5**	**0.5**
BP2: Task Composition	-	-	0.7	0.7	1	0.5	**0.85**	**0.6**
BP3: Resequencing	0.5	0.5	-	-	0	0	**0.25**	**0.25**
BP4: Parallelism	0.4	0.2	-	-	0.7	0.6	**0.55**	**0.4**
BP5: Task Automation	-	-	1	1	1	1	**1**	**1**
Average	**0.633**	**0.566**	**0.85**	**0.85**	**0.54**	**0.42**	**0.63**	**0.55**

Table 6.5: Normalized Accuracy Scores by Process and Heuristic Application

Based on the insights presented in Table 6.5, it is evident that the LLM exhibited varying degrees of effectiveness in identifying and solving optimization opportunities across the different processes and optimization opportunities. Thus several key patterns emerge when analyzing the results presented above:

In the **Design Process**, the model performed strongly, scoring 0.866 in both *Found* and *Solved* categories, particularly excelling in *Task Elimination* and *Parallelism.* This suggests that when the process structure is clear, the LLM can effectively identify and resolve optimization opportunities. In the **Implementation Process**, the accuracy dropped to 0.7, with the model performing well in *Task Automation* but struggling with *Task Composition*, revealing challenges in merging tasks. In the **Go-Live Process**, the model struggled the most, scoring 0.54 for *Found* and 0.42 for *Solved.* Its inability to identify redundant tasks in this more complex process highlights the need for additional contextual information to enhance performance in intricate scenarios.

When viewed across all processes, the **Heuristic** *Task Automation* emerged as the model's strongest heuristic, consistently applied across different contexts. In contrast, *Task Elimination* and *Task Composition* showed more variability, particularly in the more complex *Go-Live Process.* The model struggled most with *Resequencing* and *Parallelism*, resulting in the lowest overall scores among all tested heuristics.

6.2.5 Learning and Theorizing

Based on the insights, we can clearly conclude that using the bitmap format has yielded the expected results. The model is now clearly able to understand and optimize processes to a certain degree, demonstrating the validity of the adjusted approach as well as the capabilities of multimodal LLMs to identify objects in images when leveraging their internal capabilities instead of relying on other practices like OCR. We are further able to conclude that the successful application of *redesign heuristics* to optimize business processes is influenced by two key factors:

1. **The level of insight required to leverage a heuristic**: This is evident from the model's ease in applying *Task Automation* on explicitly labeled tasks (e.g., `manual testing`), which requires minimal contextual understanding. However, the model performs worse applying *Task Elimination* and *Task Composition*, as these heuristics demand a deeper grasp of task dependencies and inefficiencies. It struggles to identify redundant tasks and consolidate smaller tasks, particularly when tasks are only vaguely labeled as non-essential. The most significant challenges arise with *Parallelism* and *Resequencing*, which demand an understanding of workflow interdependencies and the information flow between steps, an area where the model struggled considerably.

2. **The complexity of the underlying process**, specifically the interactions between optimization potentials: In simpler, well-structured processes, such as the *Design Process*, the model performed better, applying automation and other heuristics

> with greater accuracy. However, in more processes where a greater and intertwined amount of optimization potentials like the *Go-Live Process*, it struggled significantly more to interpret task relationships correctly. This indicates that the LLM's difficulties are increased by both the complexity of the process and the lack of contextual insight into task interdependencies.

We theorize that adding more context to the prompt about the way in which the company operates, as well as providing the model with processes that contain more isolated and less interwoven optimization potentials, could significantly improve the model's ability to provide and enhance its accuracy in optimizing processes.

6.3 Context-Enhanced Assessment of Process Optimization Capabilities

6.3.1 Problem Identification and Contextual Understanding

As observed in previous cycles, the LLM faces significant challenges in optimizing processes with complex interdependencies and intertwined optimization potentials, particularly where it lacks sufficient context to fully comprehend task relationships. While it performs reasonably well in simpler processes where task dependencies are more explicit, its performance decreases significantly in optimizing intricate workflows that require deeper insight into operational dynamics.

To address these limitations without altering the processes themselves, ensuring comparability to the previous cycle, we aim to test whether providing the LLM with additional organizational context enhances the results of our experiments. This mirrors real-world scenarios where process optimization is more effective when the process engineer has a detailed understanding of an organization's goals, constraints, and priorities. Furthermore, it aligns with the importance of context in driving LLM performance, as discussed in section 2.2. We hypothesize that this additional input will allow the model to apply redesign heuristics more accurately, particularly in the complex scenarios where it previously performed considerably worse.

6.3.2 Action Planning

As this cycle we aim to improve the LLM's performance by enriching the prompts with additional organizational context, all aspects of the experimental setup apart from the prompts remain the same as the previous cycle. This includes keeping the best-practice standard and its guidelines, processes, and input format unchanged. By solely adjusting the context provided to the model, we can directly compare the impact that the added context has on the LLM's ability to optimize business processes.

The prompts that we use in this cycle are thus representing the core part of the adaptations of this cycle. As already outlined, we aim to give the model more information on the organization in which the process is executed similar to as it would be done in real-world conditions. Thus for all three processes, we have created an individual prompt, providing additional context on the organization, which aims to drive improved optimization results.

The first one of these prompts (Prompt 4) is written for the *Design Process*. The focus lies hereby on providing the model with an understanding of the role that correct wording and translations play in the mockups, as in the previous cycle, the model overvalued these and thus created several rework loops for the translation team, while in the actual process, translations and wordings are not defined until the design of the mockup is fixed. Furthermore, we also provide the model with context on how risk management is performed in the organization as in the previous cycle the model just performed it at the end of the design process instead of concurrently with it, which is practiced in the original process.

> Consider the following:
>
> Our mockups prioritize Human-Computer interaction, crafted mainly by UI Designers, and do not require perfect wording. Given that the translation team is in high demand and working on multiple projects simultaneously, it's crucial to minimize their workload.
>
> Management requires a risk analysis that evolves together with the design. Based on past experiences, delaying updates until the process is complete may not meet management's expectations.

Prompt 4: Process Optimization with Enhanced Context: Design Process

The second one (Prompt 5) focuses on the *Implementation Process*. Here we provide information on the *Just-in-Time Documentation* and *Code-Driven Development* practices which the organization employs with the goal to align the optimization proposal that the model provides closer with the original process as in the previous cycle, the model proposed different ways to optimize the process which did not align with these practices that the organization employs.

> Consider the following:
>
> In our organization, we prioritize efficiency by using the "Just-in-Time Documentation" approach, ensuring documentation reflects the final implementation and avoiding unnecessary rework.
>
> We also do not follow the "Test-Driven Development" strategy to reduce the risk of missing key requirements or introducing bugs due to premature test case creation.

Prompt 5: Process Optimization with Enhanced Context: Implementation Process

The third prompt (Prompt 6) focuses on the *Go-Live Process*. In this case, we aim to address the model's inability demonstrated in the previous cycle to identify the potential for *task elimination* and recognize that performing the `user satisfaction analysis` before the product is live (i.e., before the `Go Live` task) is not feasible. To address the first issue, we provide context on the organization's focus on efficiency over excessive control, while for the second one, we clarify when tickets are issued.

> Consider the following:
>
> In our organization, we prioritize efficiency by reducing team-internal controls that do not add clear value to stakeholders as well as focus on parallelizing tasks or combining them if they use the same people, simplifying our processes and their execution.
>
> Additionally, to clarify: tickets are created post-Go-Live, as feedback is only gathered once the system is live.

Prompt 6: Process Optimization with Enhanced Context: Go-Live Process

6.3.3 Action Taking

Building on LLM-interactions from the previous cycle, we then provided the model with the specified prompts, thus once again deriving $R = 5$ outputs for each adapted process model. The outputs generated in response to each prompt are presented in section 9.2.2 under the corresponding output of the previous cycle.

6.3.4 Evaluation

Below, once again, a detailed analysis of the model's responses to each process and the corresponding optimization potentials, that follows the approach detailed in subsection 4.3.1, is presented.

Design Process

- *BP1*: In the first three instances, the model correctly eliminates the redundant task. However, in the remaining two instances, the model misinterprets the superfluous `Design Check` as an optional step, with the `Project Manager` conducting less frequent reviews. Meanwhile, the `Mockup Review: Human-Computer Interaction`, overseen by the `UI Designer`, is used more frequently to guide the design process, ultimately adding, rather than reducing, control loops.

- *BP3*: The `translation team` is now involved only when the mockup is finalized or nearly complete, minimizing the redundant work previously caused by repeated translations due to ongoing mockup changes.

- *BP4*: In all instances, the risk analysis is parallelized with the remaining design steps, concluding either before or after the `freeze mockup` step, aligning with the finalization of the design.

Based on the observations and allocated points in this experiment, we observe an increase in the Accuracy Scores compared to the previous cycle. The Accuracy Score for the found

Best Practice	LLM-Output 17		LLM-Output 19		LLM-Output 21		LLM-Output 23		LLM-Output 25	
	Found	Solved	Found	Solved	Found	Solved	Found	Solved	Found	Solved
BP1	1	1	1	1	1	1	0	0	0	0
BP3	1	1	1	1	1	1	1	1	1	1
BP4	1	1	1	1	1	1	1	1	1	1

Table 6.6: Points Allocation for Found and Solved Opportunities: Design Process (Context-Enhanced)

optimization potentials, calculated with R repetitions and $I = 3$ optimization potentials, is determined using the *Allocated Points* in the *Found* column, resulting in:

$$\begin{aligned} \text{Accuracy}_{\text{Found}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Found)}_{i,r} \\ &= \frac{1}{5 \times 3} \times 13 = 0.866 \end{aligned} \tag{6.9}$$

Similarly, the Accuracy Score for the solved optimization potentials, calculated using the *Allocated Points* in the *Solved* column, yields:

$$\begin{aligned} \text{Accuracy}_{\text{Solved}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Solved)}_{i,r} \\ &= \frac{1}{5 \times 3} \times 13 = 0.866 \end{aligned} \tag{6.10}$$

Implementation Process

- *BP2:* As in the first cycle, the model correctly recognized and leveraged the *task combination* heuristic in two instances. Differently from the first cycle, however, the model did not recognize any optimization opportunity regarding the `Create Test Checklist` task in three instances, resulting in a worse overall performance on this heuristic compared to the previous cycle.
- *BP5:* As in the previous cycle, the model suggests automating tests where applicable. However, with the introduction of TDD in this cycle, the model proposes a more rigorous approach by incorporating both automated and manual testing to ensure comprehensive bug detection.

Considering these observations and point allocations, we observe a decrease in the Accuracy Scores compared to the previous cycle. The Accuracy Score for the found optimization potentials, calculated using R repetitions and $I = 2$ optimization potentials, is determined by the *Allocated Points* in the *Found* column, resulting in:

Best Practice	**LLM-Output 27**		**LLM-Output 29**		**LLM-Output 31**		**LLM-Output 33**		**LLM-Output 35**	
	Found	Solved	Found	Solved	Found	Solved	Found	Solved	Found	Solved
BP2	1	1	0	0	1	1	0	0	0	0
BP5	1	1	1	1	1	1	1	1	1	1

Table 6.7: Points Allocation for Found and Solved Opportunities: Implementation Process (Context-Enhanced)

$$\begin{aligned}\text{Accuracy}_{\text{Found}} &= \frac{1}{R \times I}\sum_{r=1}^{R}\sum_{i=1}^{I} \textit{Allocated Points (Found)}_{i,r} \\ &= \frac{1}{5 \times 2} \times 7 = 0.7\end{aligned} \tag{6.11}$$

Similarly, the Accuracy Score for the solved optimization potentials, calculated from the *Allocated Points* in the *Solved* column, yields:

$$\begin{aligned}\text{Accuracy}_{\text{Solved}} &= \frac{1}{R \times I}\sum_{r=1}^{R}\sum_{i=1}^{I} \textit{Allocated Points (Solved)}_{i,r} \\ &= \frac{1}{5 \times 2} \times 7 = 0.7\end{aligned} \tag{6.12}$$

Go-Live Process

- *BP1*: As in the previous step, the model failed to recognize the introduced superfluous step. Instead of eliminating it, the model once again merged the step with others, which also impacted the evaluation of subsequent criteria. In one instance, the model even parallelized the step, resulting in further decreased performance compared to the last cycle.

- *BP2*: As outlined above, the model's failure to correctly identify the superfluous task again leads to decreased performance in applying this heuristic, as it struggles to find an appropriate optimization. Additionally, in this cycle, the model even fails to recognize the potential to apply the heuristic in one instance, resulting in a reduced overall score compared to the previous cycle.

- *BP3*: The model correctly moves the task in all instances as intended. Notably, in two instances, the model integrates the task into the Go-Live phase as part of its effort to automate it.

- *BP4*: The model identifies the potential to parallelize the `PR Campaign` with other tasks in all instances, correctly parallelizing it with the optional `Presentation` task in three cases and validly with the `Final Test` in one case. In one instance, however, it is incorrectly parallelized with a hallucinated task called `Go-Live Readiness Review`.

- *BP5*: As in the previous cycle, the model consistently identifies and optimizes the task that can be automated in every instance.

Best Practice	**LLM-Output 37**		**LLM-Output 39**		**LLM-Output 41**		**LLM-Output 43**		**LLM-Output 45**	
	Found	Solved	Found	Solved	Found	Solved	Found	Solved	Found	Solved
BP1	0	0	0	0	0	0	0	0	0	0
BP2	1	0.5	1	0.5	0	0	1	0.5	1	0.5
BP3	1	1	1	1	1	1	1	1	1	1
BP4	1	1	1	1	1	0	1	0.5	1	1
BP5	1	1	1	1	1	1	1	1	1	1

Table 6.8: Points Allocation for Found and Solved Opportunities: Go-Live Process (Context-Enhanced)

Given the observations and point assignments, this results in higher Accuracy Scores compared to the previous cycle. The Accuracy Score for the found optimization potentials, calculated using R repetitions and $I = 5$ optimization potentials, is derived from the *Allocated Points* in the *Found* column, resulting in:

$$\begin{aligned}\text{Accuracy}_{\text{Found}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Found)}_{i,r} \\ &= \frac{1}{5 \times 5} \times 19 = 0.76\end{aligned} \tag{6.13}$$

Similarly, the Accuracy Score for the solved optimization potentials, calculated from the *Allocated Points* in the *Solved* column, yields:

$$\begin{aligned}\text{Accuracy}_{\text{Solved}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Solved)}_{i,r} \\ &= \frac{1}{5 \times 5} \times 15.5 = 0.62\end{aligned} \tag{6.14}$$

Complete Analysis

As a result of the observations and point allocations outlined above, the overall Accuracy Scores for this cycle can be calculated as follows. The Accuracy Score for the found optimization potentials, based on R repetitions and the $I = 10$ optimization potentials that were introduced across all three processes, is derived from the *Allocated Points* in the *Found* column, yielding:

$$\begin{aligned} \text{Accuracy}_{\text{Found}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Found)}_{i,r} \\ &= \frac{1}{5 \times 10} \times 39 = 0.78 \end{aligned} \tag{6.15}$$

Similarly, the Accuracy Score for the solved optimization potentials is calculated using the *Allocated Points* in the *Solved* column, resulting in:

$$\begin{aligned} \text{Accuracy}_{\text{Solved}} &= \frac{1}{R \times I} \sum_{r=1}^{R} \sum_{i=1}^{I} \textit{Allocated Points (Solved)}_{i,r} \\ &= \frac{1}{5 \times 10} \times 35.5 = 0.71 \end{aligned} \tag{6.16}$$

The insights from this cycle's analysis are summarized in the updated table below:

Best Practice	**Design Process**		**Implementation Process**		**Go-Live Process**		**Average**	
	Found	Solved	Found	Solved	Found	Solved	Found	Solved
BP1: Task Elimination	0.6	0.6	-	-	0	0	**0.3**	**0.3**
BP2: Task Composition	-	-	0.4	0.4	0.8	0.4	**0.6**	**0.4**
BP3: Resequencing	1	1	-	-	1	1	**1**	**1**
BP4: Parallelism	1	1	-	-	1	0.7	**1**	**0.85**
BP5: Task Automation	-	-	1	1	1	1	**1**	**1**
Average	**0.866**	**0.866**	**0.7**	**0.7**	**0.76**	**0.62**	**0.78**	**0.71**

Table 6.9: Normalized Accuracy Scores by Process and Heuristic Application (Context-Enhanced)

Based on the insights presented in Table 6.9, it is evident that providing additional context to the LLM leads to noticeable improvements in its ability to identify and address optimization opportunities across different processes. As in the previous cycle, several key patterns emerge in the results:

In the **Design Process**, the model performs particularly well with the added context, achieving scores of 0.866 in both the *Found* and *Solved* categories. Significant improvements are observed in *Parallelism* and *Resequencing*, while a slight decline is noted in *Task Elimination* compared to the previous cycle. In the **Implementation Process**, the model's performance declines compared to the last cycle, scoring 0.7 in both categories, down from 0.85. This reduction is primarily due to decreased accuracy in applying the *Task Composition* heuristic, which was a focus of the added context but did not yield the expected improvement. The **Go-Live Process** shows the most significant improvement, with scores of 0.76 in *Found* (up from 0.54) and 0.62 in *Solved* (up from 0.42). The largest gains are observed in the application of the *Resequencing* and *Parallelism* heuristics, where the model demonstrates considerable progress due to the enhanced contextual information.

Overall, across all processes, the **heuristic** *Resequencing* joins *Task Automation* as the model's strongest heuristic, largely due to the notable improvements in the *Go-Live Process*. *Parallelism* also performs strongly across both processes in which it is applied. Conversely, *Task Elimination* and *Task Composition* show weaker performance, with *Task Composition* even declining compared to the previous cycle, despite the additional context.

6.3.5 Learning and Theorizing

The results from this cycle further highlight the critical role that context plays in improving the LLM's ability to optimize business processes. By providing the model with more detailed organizational context, it is clear that overall performance increases, particularly in complex processes like the *Go-Live Process*. However, this cycle also demonstrates that adding context is a double-edged sword, where its effectiveness is highly dependent on how it guides the model towards specific optimization decisions.

The increased performance in applying *Parallelism* and *Resequencing* in the *Design* and *Go-Live* processes shows the clear benefits of providing more organizational context. The model is better able to understand task relationships and dependencies, resulting in more precise and context-appropriate optimizations.

However, it is evident that steering the model with added context can potentially push it too strongly in specific directions and mislead the model. For instance, in the *Implementation Process*, the focus on organizational practices like Just-in-Time Documentation caused the model to overlook alternative heuristics, such as *parallelization*, which it had successfully identified in the previous cycle. Similarly, in the application of *Task Elimination* and *Task Composition*, the additional context caused the model to focus too heavily on certain aspects of the process, resulting in decreased accuracy. Furthermore, the added context sometimes led the model to contradict itself in its outputs, producing optimized process flows that do not always fully align with the explanations provided at the beginning of the output.

These findings highlight that while additional context can significantly enhance the model's performance, particularly in complex processes and for the application of heuristics that require a deeper understanding of task dependencies, it is equally important to ensure that this context does not unintentionally constrain the model's flexibility or drive it away from exploring other valid optimization pathways.

CHAPTER 7

Discussion

This chapter presents the findings from our five Action Cycles, analyzing their outcomes in relation to the existing scientific literature, while also discussing the limitations of our study. Additionally, we propose future research directions based on these results, highlighting areas for further exploration and potential advancements in the field.

7.1 Analysis of Results

7.1.1 Critical Factors influencing Accuracy

In this section, we discuss the most critical factors identified across all five cycles of our study that significantly impact the accuracy with which LLMs are able to perform BPO tasks.

The Input Format proved to be a critical factor in LLM performance during our study, particularly in the third cycle when the model struggled to interpret process flows. This issue arose because when provided with a PDF containing vector-based diagrams, most platforms (in our case *ChatGPT*) rely on OCR to extract content from it. While OCR is able to correctly identify individual elements, it often fails to maintain their proper order and placement. In earlier cycles, this issue did not surface because in them, the focus was on the syntax of individual elements rather than their semantic connections. The tasks did not depend on understanding interactions between elements, so the shortcomings of OCR were not apparent. However, when we shifted the focus to the semantic aspects of process optimization, the input format's limitations became evident, highlighting the importance of providing inputs in formats that align with the model's capabilities. Switching to the bitmap format in following cycles resolved this issue. Unlike vector graphics, bitmaps store data as a grid of pixels, enabling the model to directly interpret visual patterns. This adjustment provided the model with a accurate understanding of the process flow, leading to more accurate optimization proposals and highlighting the importance of choosing input formats that leverage the model's strengths. It also underscores the user's responsibility to prepare data in ways that allow LLMs to perform optimally.

Explicit Guidance and Context emerged as another key driver of the accuracy, particularly in the second and fifth cycles. In the first cycle, with minimal context, the model achieved a moderate accuracy (0.68) in identifying simple violations but struggled with more complex guidelines requiring deeper semantic understanding, suggesting that without sufficient context, the model can only manage tasks driven by straightforward patterns. When *goal-specific context* was introduced in the second cycle, the accuracy improved to 0.9, showing that structured guidance helps the model handle more complex process criteria, though more abstract tasks still presented a challenge to the model. A similar improvement was observed in the fifth cycle, where *domain-specific context* raised the accuracy to 0.78 in identifying optimization potentials (compared to 0.63 in the fourth cycle) and 0.71 in solving them (up from 0.55). These improvements highlight the strong influence that explicit guidance and detailed context have on driving LLM effectiveness by aligning outputs with user-intentions and organizational priorities.

Overfitting and the Challenge of Contextual Guidance became a risk that has to be considered, based on the results of the fifth cycle. While the addition of *domain-specific context* significantly improved accuracy across all metrics compared to the fourth cycle, it however also lead the model to sometimes focus too heavily on specific optimization paths, neglecting other valid strategies it had considered in the less contextualized fourth cycle, ultimately limiting its performance. This underscores the need to balance the amount of context provided to the model. While context enhances performance by aligning the model's outputs with real-world priorities, too much guidance can limit flexibility and exploration of alternative solutions, stifling the potential that the model truly entails.

Reasoning and Complexity Handling remained a persistent challenge for the model throughout all cycles, particularly in tasks involving interdependent elements. Across all cycles, despite improvements from input format changes and the introduction of more context, the model consistently struggled to manage the intricate relationships between tasks. In earlier cycles, the model was able to handle simple, isolated issues, but as the complexity of the processes increased, it became clear that the model's reasoning capabilities were insufficient for tasks that required an understanding of the interactions between multiple elements. Even in the later cycles, where context was enhanced and the input format optimized, the model still faced difficulties with higher-order dependencies and dynamic relationships within processes. This recurring issue across all cycles, despite adjustments in prompts and input formats, suggests that the model is not able to handle multi-layered, complex scenarios. We account this to the fact that we did deliberately not use any advanced prompting techniques like chain-of-thoughts, or multi-agent-reasoning (see subsection 4.2.2). This highlights an ongoing limitation in the ability of current model's to fully process and optimize more intricate processes, indicating a critical area of demand for more advanced LLM capabilities for real-world applications.

7.1.2 Answering the Specific Research Questions

Together with these insights on the main performance drivers, we are now also able to answer the three research questions which we have formulated in chapter 2, based on the analyzed existing literature, in order to guide our work more effectively and align it stringently with the requirements posed by the current SOTA in research.

Answer to SRQ1 *Are LLMs adhering to business process modeling criteria from the scientific literature?*

Current SOTA LLMs can adhere to business process modeling criteria when provided with clear, structured prompts. However, their performance is limited with more complex and abstract criteria, such as labeling and task interdependencies. While useful for less complex tasks, LLMs still require significant user input, meaning full automation of process modeling is not yet achievable.

Answer to SRQ2 *Are LLMs correctly applying optimization criteria from the scientific literature when asked to optimize a process?*

Current SOTA LLMs can autonomously apply basic optimization criteria effectively but struggle with complex tasks, such as parallelism or task elimination, particularly in interdependent processes. While they show potential in automating simpler optimizations, without some amount of fine-tuning, they are not capable of fully automating the optimization of more intricate processes according to an organizations standards without human guidance.

Answer to SRQ3 *What specific limitations and challenges are identified in using LLMs for BPM-optimization tasks?*

Key limitations include challenges in processing complex input formats, difficulty reasoning through interdependent tasks, and a dependence on explicit prompts to guide the model. Additionally, providing too much context led to overfitting, emphasizing the need for a careful balance in the amount of context provided.

7.2 Relation and Contributions to Existing Literature

Our research relates on existing work that explores LLMs-capabilites for BPM-lifecycle tasks, by building on the body of existing work while extending on notable gaps or less explored aspects in several ways.

The technical main findings of our work regarding the approach that should be taken, which are presented above (see section 7.1), are related to the existing literature in the following way:

- **Input format** emerged as a critical factor in our research, expanding prior findings that emphasize the importance of high-quality data for automation as well as the SOTA research on multimodal LLMs [DRMR13, Ope23a]. While earlier works assumed structured inputs, we utilized unstructured formats to reflect real-world conditions, revealing LLM limitations in dependence of the input format and offering insights into the intricacies of using multimodal LLMs for BPO.

- **The impact of context** also proved to be a critical aspect in our research. As highlighted in previous studies [ZZPC23], context plays a vital role in LLM performance. We incorporated this factor into two of our five action cycles, leading overall to significantly improved results. However, these improvements did not come without drawbacks.

- **Overfitting** presented a significant challenge in our study when in-depth context was provided to the model. While prior research on LLMs has underscored the significance of this aspect, with Vidgof et al. [VBM23b] discussing the rigidity of LLM outputs when constrained by specific prompts in the context of BPM, our findings extend this understanding by demonstrating how excessive domain-specific context can further limit model flexibility. This suggests the need for balancing explicit guidance with adaptability to prevent narrowing the model's focus on optimization tasks.

- **Reasoning and complexity handling** remain problematic for LLMs, especially when dealing with task interdependencies and dynamic relationships. Previous works, such as Grohs et al. [GAER23], have focused on simpler, structured tasks, while our research highlights the limitations LLMs face in fully optimizing more intricate, multi-layered processes with various degrees of complexity.

Methodologically our study aligns with existing research utilizing the AR framework to evaluate LLMs, such as the work by Li et al. [LZZ+24]. By applying this well-established method in a relatively novel context, we not only adhere to scientific practices but also reinforce their validity by demonstrating their feasibility in a new domain. Additionally, the evaluation framework developed for this work, which incorporates best practices and a quantitative scoring system, offers a replicable and scalable approach for assessing LLMs in less easily quantifiable domains. This framework can be applied not only in the further research directions outlined in section 7.4, but also in other studies facing similar challenges, addressing the current gap in empirical, reproducible evaluation techniques for LLM-driven research.

7.3 Limitations of our work

This study, like all research, is subject to certain limitations that impact the scope and generalizability of its findings. These limitations, which we discuss below in detail, provide important context for interpreting the results and highlight areas for future research (see section 7.4).

7.3.1 Limitations in Scope

The **limited process scope** stems from the study's focus on a single process and its subprocesses. While this approach enabled a thorough analysis of the model's capabilities within the specific context of digitization projects, it limits the generalizability of the findings to other types of real-world processes (e.g., production or logistics workflows). Additionally, the process in focus was provided by a single source, which may not fully represent the variability or complexity encountered across different sources. This scope restricts the ability to assess how the model might perform when dealing with a broader range of processes provided from different origins.

Furthermore, **limited heuristics scope** also affects the study's generalizability, as we followed only one standard each for the semantic and syntactic analysis of model capabilities in working with processes. Testing the capabilities on the basis of different best practices could yield different results, potentially offering a more comprehensive understanding of how well the model handles various redesign strategies.

Another potential limitation is presented by the **limited model scope**. Our exclusive focus on *GPT-4o*, although a leading SOTA model, narrows the scope by not including other prominent models such as *LLama* or *Gemini*, which may perform differently. Furthermore, during the course of writing this work, previews of more advanced models and interaction methods, such as *GPT-4o1* and *ChatGPT Canvas*, were released, offering enhanced capabilities that could lead to significantly improved results in future studies.

7.3.2 Potential Biases

Bias inherent to Action Research presents a further limitation in our study, as the researcher's direct interaction with the entity being analyzed may introduce unintentional biases. This dynamic can influence the model's responses, as seen in the fifth cycle, where the model overfitted to specific optimization strategies.

Additionally, **bias in model behavior**, caused by the hidden system prompt that shapes the general behavior of models like those used in *ChatGPT*, may have influenced the model's responses, aligning them with preconfigured patterns not explicitly directed by user prompts [Mic24a]. While this limitation could reduce the model's flexibility, using *ChatGPT* was essential for maintaining real-world applicability, as most users interact with LLMs through platforms like *ChatGPT* or *MS Copilot*, both of which rely on system prompts to guide outputs.

7.4 Future Research Directions

While our study highlights the potential of LLMs in BPO, several areas, which are outlined below, remain unexplored, presenting important directions for future research.

7.4.1 Refined prompting

Our study revealed, that a refined prompting approach is crucial to drive results which align with each organizations needs while leveraging with the full capabilities that current models entail. This leads to our call towards a refined prompting approach specifically for the optimization of BPO. Since we have shown that models are very much capable of optimizing business processes, the next step is to improve on our results by improving on our minimalistic prompts. Hereby we call for future research to develop rules that can be applied by practitioners on how much context should be given to the model and how the model could be guided to follow best-practices while being creative in it's approaches and following an organizations practices.

7.4.2 Capabilities of Next-Generation Models

Next-generation models like *GPT-4o1* and new interaction methods such as *ChatGPT Canvas* introduce enhanced reasoning abilities and collaborative features, making them well-suited to handle more complex processes. These models will require a re-evaluation of our findings as they become fully available, updating conclusions to reflect their advanced capabilities.
Another crucial aspect for future research is the potential of models embedded within organizational ecosystems, such as *Microsoft Copilot*, which allows models to directly access organizational data. Future studies should examine how these integrations can enhance the alignment of model outputs with evolving business goals, while addressing the challenge of preventing overfitting.

7.4.3 Leveraging LLMs to enhance BPM

Throughout our study, it became clear that LLMs have the potential not only to support existing practices but also to significantly enhance them. One of the most promising directions for future research lies in the real-time generation of process abstractions tailored to different users. The complexity of business processes often makes them difficult for all members of an organization to understand, or they may feel stifling and overly restrictive. A solution could involve dynamically simplifying processes based on the user's role or needs, with LLMs generating customized, real-time views of processes. This would eliminate the need to store multiple versions, while maintaining a detailed version within the organizational ecosystem. Such an approach could make processes more accessible while preserving their complexity where necessary. Future research should explore the feasibility of this method, focusing on developing rules to maintain critical interdependencies, avoid oversimplification, and maximize usability.

CHAPTER 8

Conclusion

8.1 Summary of Findings

Based on the insights gained through the five action cycles of this work, and the analysis made in chapter 7, we are now able to answer our overarching research question (ORQ), which we formulated in chapter 1 in order to formally close the identified research gap:

Answer to the ORQ *Can Large Language Models be leveraged to automate business process optimization, thereby overcoming the limitations of current automation methods?*

Current SOTA LLMs can automate BPO to a certain degree but still require human input for more complex aspects, like handling task interdependencies and intricate process flows. While they improve efficiency, challenges like overfitting and difficulties with complex inputs highlight the need for careful guidance and oversight. This suggests that full automation of BPO remains unachievable without further models developments, a deep integration into organizational systems to provide the necessary context, a carefully crafted approach and prompt design, or a combination of them.

8.2 Implications to the Field

This study highlights important implications for both the academic and practical applications of LLMs in BPO

From an **academic perspective**, our findings suggest that LLMs hold potential for BPM beyond more quantifiable tasks like process mining and -monitoring, as we were able to shown their potential in optimization, addressing pressing research questions, as described in section 7.2. However, their limitations in managing complex, interdependent processes underscore the need for further studies in the area, pointing towards a critical area for future research, with several of them having been outlined in section 7.4.

On a **practical level**, our research demonstrates that LLMs can significantly reduce the time and resources required for BPO tasks, though full automation of complex processes remains unattainable at present. As the technology continues to advance, LLMs are poised to become increasingly valuable in supporting process optimization and streamlining operations. However, businesses must recognize the current limitations of LLMs, particularly the importance of precise prompt design and the necessity of ongoing human oversight for managing more complex tasks. Rather than replacing human expertise, LLMs should be viewed as powerful augmentation tools, enhancing optimization efforts while still relying on human operators to mitigate risks such as overfitting and handling task interdependencies. Therefore, we call on the industry to capitalize on the capabilities these models offer today while preparing for future iterations that will undoubtedly further disrupt and transform the way we work.

8.3 Final Thoughts

As LLMs continue to evolve, their role in BPO will likely expand, though full automation remains beyond reach at this time. This research has demonstrated that while LLMs can effectively assist with simpler tasks, they are not yet capable of autonomously optimizing processes. While limitations such as managing complex interdependencies and avoiding overfitting necessitate human oversight, LLMs already show considerable potential in augmenting BPO efforts. Organizations should view LLMs as tools that enhance efficiency rather than replace human expertise, using them to streamline tasks while maintaining control over more intricate processes.

An important point we want to reiterate is that LLMs, much like employees, require clear guidance to perform effectively. They are not all-knowing AI systems, but rather highly advanced predictive engines that generate output based on the input and context they receive. Therefore, just as employees need clear direction, LLMs must be carefully instructed with precise prompts and context to achieve optimal results.

Looking ahead, as LLMs continue to evolve and become increasingly capable, their ability to autonomously manage complex tasks is likely to grow. However, realizing this potential and unlocking their full value will require ongoing research and strategic integration into business workflows, ensuring that these models are leveraged effectively while being carefully guided and monitored.

CHAPTER 9

Appendix

Contents

9.1 Appendix to Chapter 5 70

9.1.1 Process Models 70

9.1.2 Model Outputs 72

9.2 Appendix to Chapter 6 85

9.2.1 Process Models 85

Design 85

Implementation 87

Go-Live 89

9.2.2 Model Outputs 91

PDF 91

Design Process 91

Bitmap 96

Design Process 96

Implementation Process 110

Go-Live Process 123

9.1 Appendix to Chapter 5

9.1.1 Process Models

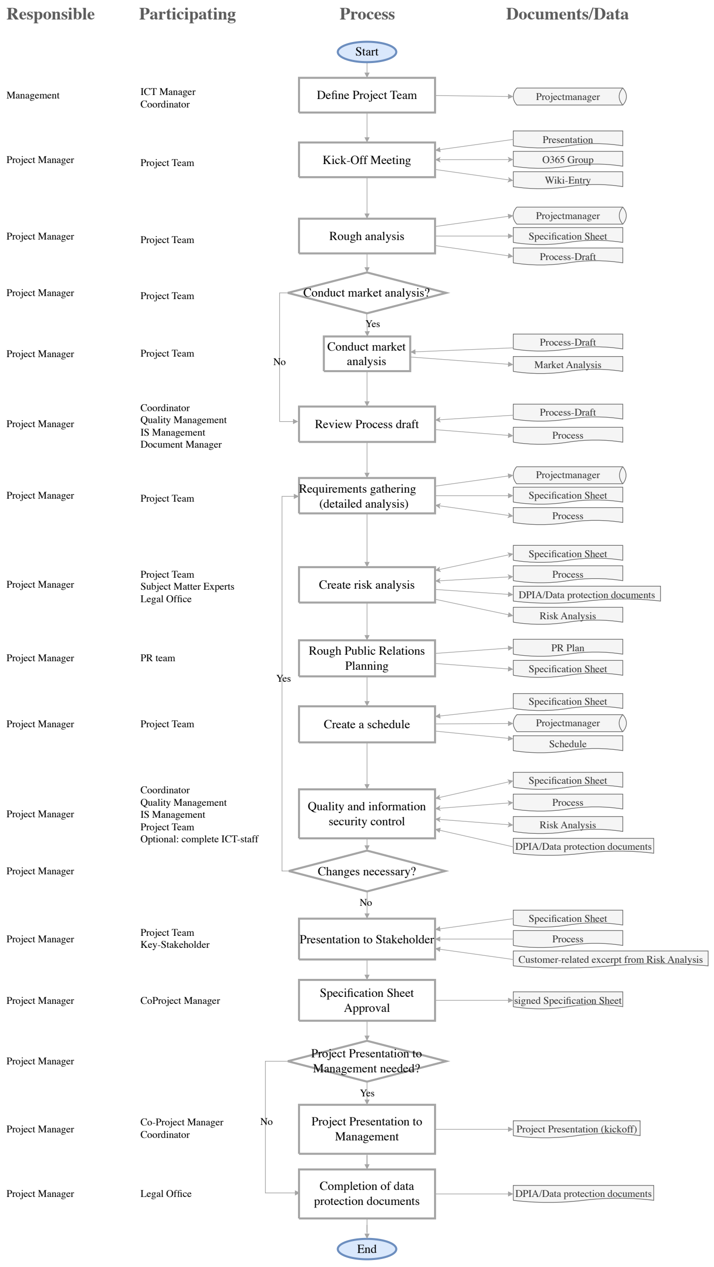

Figure 9.1: Original Analysis Process

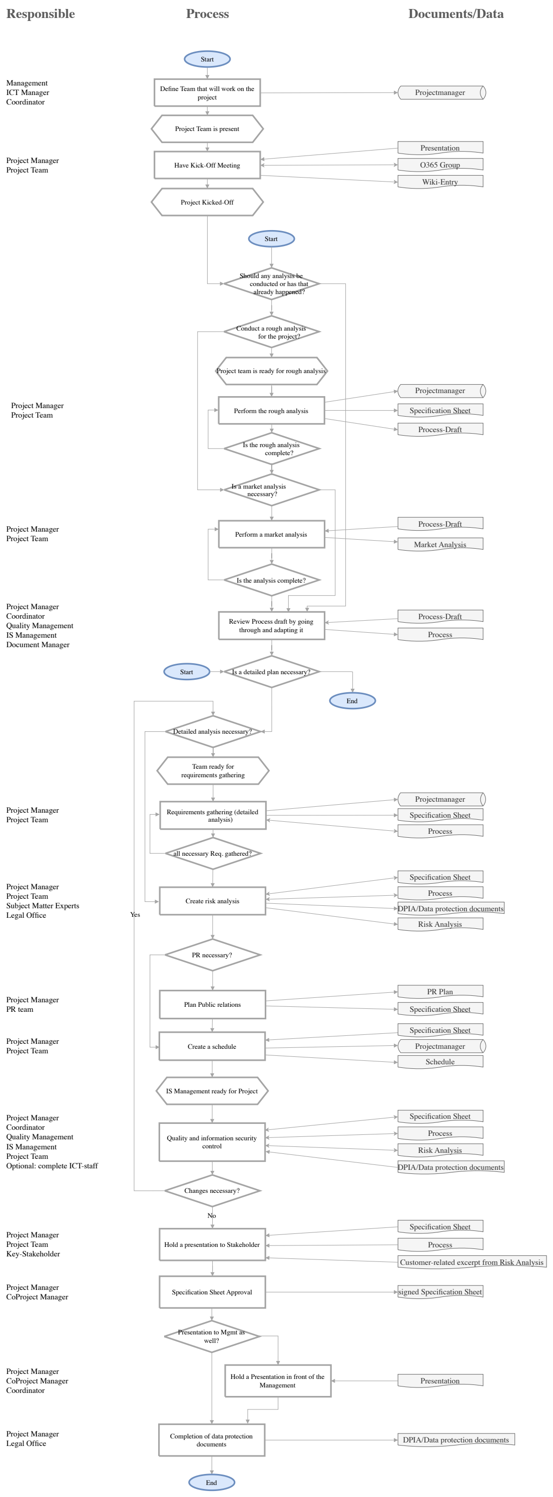

Figure 9.2: Adapted Analysis Process

9.1.2 Model Outputs

R1

Based on the analysis of the attached process model, here are several observations highlighting elements that do not adhere to best-practice modeling standards, along with areas for improvement:

1. **Inconsistencies in Symbol Usage**:
   - Symbols are used inconsistently, such as the differentiation between decision gateways (diamonds) and activities (rectangles). This leads to confusion regarding the flow of decisions versus actions.
2. **Redundant Gateways**:
   - Several decision gateways seem redundant, such as multiple checks asking if an analysis or review is necessary. These could be combined or simplified to streamline the process and avoid unnecessary complexity.
3. **Unclear Connections**:
   - Connections between some tasks and gateways lack clarity. For instance, it's not always clear how decision outcomes flow into subsequent activities. This lack of directional clarity could confuse the reader about the correct sequence of steps.
4. **Excessive Complexity**:
   - The process model includes several parallel and nested decisions, making the flow appear overly complex. Simplifying decision paths and reducing the number of sequential decisions could enhance readability and reduce cognitive load.
5. **Misuse of Symbols**:
   - Symbols like start and end events are not consistently used, leading to unclear process initiation and termination points. Additionally, there are elements like "Yes" and "No" labels that are floating without clear visual connection to corresponding decision points, making the process difficult to follow.
6. **Deviations from Standard Modeling Conventions**:
   - The model sometimes deviates from standard BPMN (Business Process Model and Notation) conventions. For instance, tasks and decisions are not always correctly labeled or aligned with the expected BPMN elements, making it less intuitive for users familiar with these standards.
7. **Ambiguous Task Descriptions**:
   - Some task descriptions are vague or overly generic, such as “Perform a rough analysis” or “Create a schedule,” without specifying the input or output artifacts. This ambiguity could lead to misinterpretation of what is required at each step.
8. **Lack of Clear Roles and Responsibilities**:
   - Although roles are listed alongside tasks, there is inconsistency in how responsibilities are assigned. Some tasks list multiple roles without specifying who is primarily responsible, potentially leading to confusion during implementation.
9. **Missing Process Documentation or References**:
   - While some tasks reference specific documents (e.g., Specification Sheet, Risk Analysis), others do not, which creates gaps in understanding what inputs are required or outputs expected. Consistent documentation references would improve clarity.
10. **Gateways without Defined Outcomes**:
    - Several gateways lead directly into tasks without clearly defined “Yes” or “No” outcomes, which can mislead the process flow. This should be corrected by ensuring that each decision point clearly delineates possible paths.

Recommendations for Improvement

- **Simplify Decision Paths**: Reduce the number of decision gateways and consolidate similar checks to streamline the flow.
- **Consistent Use of Symbols**: Adhere to standard BPMN symbols for tasks, gateways, events, and connectors to enhance readability and maintain consistency.
- **Clarify Connections and Outcomes**: Ensure all connections between tasks and gateways are clearly defined, with directional arrows showing the exact flow of the process.
- **Define Roles Clearly**: Specify primary responsibility for each task to avoid ambiguity in execution.
- **Label Outcomes Explicitly**: Clearly label all outcomes of decision gateways to guide the user through the process without confusion.
- **Enhance Task Descriptions**: Provide clearer task descriptions, detailing what inputs are needed and what outputs are expected.

These adjustments would significantly enhance the clarity, simplicity, and overall effectiveness of the process model, aligning it with best-practice modeling standards.

LLM-Output 1: Assessment of Process Model Quality R1

The Seven Process Modeling Guidelines (7PMG) provide a framework to ensure clarity, consistency, and quality in process modeling. Here's an evaluation of the provided process model against each of the 7PMG guidelines, rated on a scale of 1-10:

1. **Use as Few Elements as Possible (Simplicity)**
   - **Rating:** 4/10
   - **Reasoning:** The model uses numerous decision gateways, some of which seem redundant and unnecessary. There is a high level of complexity with multiple decision points that could be simplified. This makes the model more complicated than it needs to be.
2. **Minimize the Routing Paths per Element (Structuredness)**
   - **Rating:** 3/10
   - **Reasoning:** Many tasks are linked with complex routing paths that create confusion. The decision points often lead to multiple paths, making the flow difficult to follow. Better structuring with fewer and clearer paths would improve adherence to this guideline.
3. **Use One Start and One End Event (Clarity)**
   - **Rating:** 5/10
   - **Reasoning:** The model has multiple starting and ending points, making it unclear where the process truly begins and ends. A single start and end event would greatly improve the overall clarity and provide a clear boundary for the process.
4. **Model as Structured as Possible (Structured Modeling)**
   - **Rating:** 4/10
   - **Reasoning:** The model's structure is inconsistent, with unstructured decision-making and unclear flows between tasks. The nesting of decisions and parallel flows creates a lack of coherence, making it hard to follow a single logical path through the process.
5. **Avoid OR-Gateways as much as possible (Model Simplicity)**
   - **Rating:** 6/10

- **Reasoning:** Although the model primarily uses simple decision points (Yes/No), the frequency and arrangement of these decisions are excessive, leading to unnecessary complexity. Simplifying or merging similar decisions would enhance this aspect.

6. **Use as Few Different Types of Elements as Possible (Uniformity)**
   - **Rating:** 6/10
   - **Reasoning:** The model uses a limited range of element types, but the usage is inconsistent, and the roles and symbols are not always clearly differentiated. This inconsistency detracts from the model's overall uniformity and interpretability.
7. **Decompose a Model with More than 50 Elements (Modularity)**
   - **Rating:** 3/10
   - **Reasoning:** The process contains a high number of elements without clear modular decomposition. Breaking down the process into sub-processes or modules would significantly enhance readability and manageability, aligning it better with this guideline.

Overall Assessment:

- **Total Average Score:** 4.4/10

Summary: The process model shows poor to moderate adherence to the Seven Process Modeling Guidelines. The primary areas needing improvement include reducing complexity by simplifying paths, structuring the model more clearly, and adhering to a single start and end event. To improve, the model should focus on restructuring decision points, reducing redundant paths, and breaking down complex sections into simpler, modular components.

LLM-Output 2: Context-enhanced Assessment of Process Model Quality R1

R2

After analyzing the provided process model, I identified several areas where the modeling does not adhere to best practices and could be improved for clarity, simplicity, and accuracy:

1. **Inconsistencies and Redundant Gateways**
   - **Multiple Decision Points:** The model includes several decision points (represented by diamonds) that could be redundant or unnecessary. For example, questions like "Is a market analysis necessary?" and "Should any analysis be conducted or has that already happened?" could potentially be combined into a single decision point to simplify the model.
   - **Excessive Use of Gateways:** The frequent use of gateways without clear divergence or convergence points can make the process difficult to follow. This includes instances where a simple sequence flow could replace a gateway.
2. **Unclear Connections**
   - **Ambiguous Flow:** In some cases, it is unclear which process step is connected to which decision point or activity. This lack of clarity in the sequence flows can lead to confusion about the process order and dependencies.
   - **Unlabeled Arrows:** Some arrows in the process model are not labeled, which leaves the nature of the transitions between activities unclear. Labeling the flows, especially where decisions are involved, can provide much-needed clarity.
3. **Misused Symbols**
   - **Incorrect Use of Gateways:** Gateways are used where simple flow connectors would suffice, particularly in situations where there is no actual branching or merging of paths. This misuse can confuse the viewer regarding the process's complexity.
   - **Symbol Choice:** The symbols used for activities and decisions are sometimes inconsistent or do not clearly represent the intended action, leading to potential misinterpretation.

4. **Excessive Complexity**
   - **Over-complicated Process Flows:** Some parts of the model include a level of detail that may not be necessary at this stage of the process, leading to excessive complexity. Simplifying these areas can make the model more accessible and easier to understand.
   - **Repeated Activities:** There are several instances where the same activity appears to be repeated without clear justification. This repetition can create confusion and make the model appear more complex than it needs to be.
5. **Deviations from Standard Modeling Conventions**
   - **Start and End Events:** The use of start and end events is inconsistent. Some paths in the process flow seem to lack a clear end, which can lead to ambiguity about when or how the process concludes.
   - **Role and Responsibility Assignments:** The assignment of roles (like Project Manager, Coordinator, etc.) to specific tasks is not always consistent. This inconsistency can cause confusion about who is responsible for each part of the process.

Recommendations for Improvement:

- **Simplify the Decision Points:** Combine similar decision points where possible to reduce the complexity of the model.
- **Clarify Flow Connections:** Ensure that all arrows are clearly labeled and that each flow is logical and easy to follow.
- **Use Symbols Correctly:** Re-evaluate the use of gateways and other symbols to ensure they accurately represent the intended process logic.
- **Reduce Redundancy:** Eliminate unnecessary repetition of activities and decision points to streamline the model.
- **Standardize Role Assignments:** Make sure that roles are consistently assigned to tasks throughout the model.

These changes would improve the overall readability and correctness of the process model, making it more aligned with best practices in process modeling.

LLM-Output 3: Assessment of Process Model Quality R2

The Seven Process Modeling Guidelines (7PMG) are designed to ensure that process models are clear, simple, and effective. The guidelines are as follows:

1. **G1: Use as few elements in the model as possible.**
2. **G2: Minimize the routing paths per element.**
3. **G3: Use one start and one end event.**
4. **G4: Model as structured as possible.**
5. **G5: Avoid OR gateways.**
6. **G6: Use verb-object activity labels.**
7. **G7: Decompose a model with more than 30 elements.**

Here's the evaluation of the provided process model according to these guidelines:

G1: Use as few elements in the model as possible.

- **Rating: 4/10**

- The model contains numerous decision points, activities, and roles, many of which seem redundant or could be combined to reduce complexity. There are multiple elements representing similar decisions or actions, suggesting that the number of elements could be significantly reduced.

G2: Minimize the routing paths per element.

- **Rating: 3/10**
- The model includes many elements with multiple incoming and outgoing paths, which increases the complexity and reduces clarity. For example, some decision points have several branches that could potentially be consolidated, or the process could be simplified by reducing the number of routes per element.

G3: Use one start and one end event.

- **Rating: 2/10**
- There are multiple start and end points throughout the model, which violates this guideline. The lack of a single start and end event creates ambiguity about when the process begins and ends and which flows are part of the main process versus sub-processes.

G4: Model as structured as possible.

- **Rating: 4/10**
- The model lacks clear structuring. There are numerous branching points and paths that make the process flow appear disorganized. Additionally, some paths seem to have no clear purpose or destination, which affects the overall structure and readability of the model.

G5: Avoid OR gateways.

- **Rating: 5/10**
- The model contains a mix of gateways, including several that appear to function like OR gateways, where the process can follow multiple paths based on conditions. This is acceptable in some cases but should generally be avoided in favor of simpler constructs, such as XOR or AND gateways, to ensure clarity.

G6: Use verb-object activity labels.

- **Rating: 6/10**
- Many of the activity labels in the model do follow the verb-object format (e.g., "Perform market analysis," "Create risk analysis"), but there are some instances where the labels are ambiguous or not action-oriented (e.g., "PR necessary?"). Consistency in using clear, descriptive verb-object labels is essential for readability.

G7: Decompose a model with more than 30 elements.

- **Rating: 3/10**
- The model appears to contain more than 30 elements without any visible decomposition into sub-processes. This makes the model overly complex and challenging to understand as a single view. Decomposing the model into smaller, more manageable sub-processes would enhance clarity and adherence to this guideline.

Overall Assessment:

- The provided process model does not adhere well to the Seven Process Modeling Guidelines. The overall ratings suggest that there are significant opportunities to simplify the model, reduce the number of elements, improve structuring, and clarify the process flow.

Summary of Ratings:

1. **G1: Use as few elements as possible - 4/10**
2. **G2: Minimize routing paths per element - 3/10**
3. **G3: Use one start and one end event - 2/10**
4. **G4: Model as structured as possible - 4/10**
5. **G5: Avoid OR gateways - 5/10**
6. **G6: Use verb-object activity labels - 6/10**
7. **G7: Decompose a model with more than 30 elements - 3/10**

The overall rating indicates that the process model could greatly benefit from simplification, clearer structuring, and better adherence to the 7PMG guidelines to enhance clarity, simplicity, and accuracy.

LLM-Output 4: Context-enhanced Assessment of Process Model Quality R2

R3

To analyze the correctness of the modeling in the provided process model, I will focus on identifying elements that do not adhere to best-practice modeling standards. Below are the main areas of concern:

1. **Inconsistent Use of Gateways**

   - **Redundant Gateways:** There are instances where gateways are used unnecessarily. For example, the decision points such as "Is a market analysis necessary?" or "Is a detailed plan necessary?" seem to have yes/no outcomes that are directly leading to specific tasks. If there are only two options (yes/no), and each path directly leads to a single activity, the gateway could be redundant.
   - **Unclear Gateway Conditions:** Some gateways do not have clear conditions or criteria for their paths, which can lead to ambiguity. For example, the gateway asking, "Is the analysis complete?" lacks a clear description of what constitutes 'complete.' A better practice would involve defining the completion criteria or adding annotations to clarify the decision-making process.

2. **Excessive Complexity**

   - **Overuse of Symbols and Elements:** The model contains a significant number of elements, including various actors, documents, and process steps. This can make the model appear cluttered and harder to understand. Simplifying the flow by consolidating steps where possible or using subprocesses to encapsulate related activities can improve readability.
   - **Too Many Roles in Single Tasks:** There are many tasks that list numerous roles (e.g., "Project Manager, Coordinator, Quality Management, IS Management, Project Team") without clearly indicating their specific responsibilities. This can make it unclear who is responsible for what. It would be better to assign a primary responsible role and indicate other roles as supporting actors.

3. **Unclear Connections**

   - **Ambiguous Start and End Points:** The model appears to have multiple start and end points that could create confusion. Standard practice is to have a single, clear start and end point for the entire process, or to use clearly labeled subprocesses that feed into and out of the main flow.
   - **Unlabeled or Vague Arrows:** There are several connecting arrows without clear labels or indications of what they represent. For instance, the transition between tasks like "Create risk analysis" and "Create a schedule" lacks context on why one follows the other. Adding annotations or labels to the connectors can help clarify the flow.

4. **Misuse of Symbols**

   - **Incorrect Use of Tasks and Events:** There seems to be a mix of tasks and events that are not clearly distinguished. For example, "Hold a Presentation in front of the Management" is modeled similarly to more procedural steps like "Perform the rough analysis." Events (like presentations) and tasks (like analysis) should be clearly differentiated using appropriate symbols.
   - **Improper Use of Subprocess Indicators:** In the provided model, there are complex activities (e.g., "Perform a market analysis") that could benefit from being represented as subprocesses. This would reduce clutter and improve understanding by indicating that these tasks involve multiple steps.

5. **Deviation from Standard Modeling Conventions**

   - **Non-standard Terminology:** Some elements use non-standard terminology, like "IS Management ready for Project," which may not be immediately clear to all stakeholders. It's better to use standardized terms or provide a glossary for clarity.
   - **Inconsistent Formatting of Activities:** Activities are described with inconsistent wording, such as using imperative verbs ("Create risk analysis") versus descriptive phrases ("Risk Analysis"). Standardizing the naming convention for activities improves readability.

6. **Recommendations for Improvement**

   - **Simplify the Model:** Combine or collapse steps where possible, use subprocesses to encapsulate detailed processes, and reduce the number of roles involved in each step.
   - **Clarify Decision Points and Gateways:** Ensure that all gateways are necessary and have clear labels or annotations describing the decision criteria.
   - **Standardize Symbols and Terminology:** Use BPMN or another standardized notation consistently throughout the model. Ensure that events, tasks, gateways, and other elements are clearly differentiated.

- **Improve Layout and Flow:** Arrange the model in a way that minimizes crossing lines and clusters related activities together to enhance readability.

By addressing these issues, the process model can be made more accurate, clear, and aligned with best practices, ultimately improving its usability and effectiveness for all stakeholders involved.

LLM-Output 5: Assessment of Process Model Quality R3

To evaluate the provided process model against the Seven Process Modeling Guidelines (7PMG), I will rate each guideline on a scale of 1 to 10. Here's the assessment based on the guidelines:

1. **G1: Use as Few Elements in the Model as Possible (Simplicity)**
   - **Rating: 5/10**
   - **Reason:** The process model uses many elements, including various tasks, gateways, roles, and documents, leading to a relatively complex representation. While some complexity may be inherent to the process, there are opportunities to reduce the number of elements by consolidating tasks, using subprocesses, and removing redundant gateways.
2. **G2: Minimize the Routing Paths per Element (Cognitive Effectiveness)**
   - **Rating: 4/10**
   - **Reason:** There are multiple decision points with several outgoing paths, leading to a complex routing structure. Some gateways have multiple routes without clear distinctions or labels, making it harder for the reader to understand the flow at a glance. Reducing the number of outgoing paths per element would improve clarity and adherence to this guideline.
3. **G3: Use One Start and One End Event (Clarity)**
   - **Rating: 3/10**
   - **Reason:** The model features multiple start and end points, which can cause confusion. Standard best practices suggest a single start and end point for overall clarity. The current model's use of multiple start and end points can make it difficult to discern where the process begins and ends.
4. **G4: Model as Structured as Possible (Structuredness)**
   - **Rating: 5/10**
   - **Reason:** While some portions of the process model are reasonably structured, there are sections where the flow appears erratic or disorganized. For example, the decision points are not always logically structured, and the flow between activities could be better aligned to follow a more straightforward pattern. Enhancing the process's structured nature would improve its readability and logical consistency.
5. **G5: Avoid OR-Joins (Precision)**
   - **Rating: 7/10**
   - **Reason:** The model appears to avoid the use of OR-joins, which is in line with best practices. However, there are still points where the decision-making logic could be clearer, such as simplifying complex decisions into more straightforward binary choices (yes/no). This guideline is mostly adhered to, but improvements could be made by clarifying and simplifying decision points further.
6. **G6: Use Verb-Object Activity Labels (Consistency)**
   - **Rating: 6/10**
   - **Reason:** The model inconsistently uses verb-object labels for activities. While some activities follow the "Verb-Object" format (e.g., "Create risk analysis"), others are less clear or formatted differently (e.g., "Risk Analysis" without a verb). Ensuring consistent use of the "Verb-Object" format would improve the clarity and uniformity of the process.
7. **G7: Decompose the Model If It Has More Than 50 Elements (Decomposition)**

- **Rating: 4/10**
- **Reason:** The model appears to have more than 50 elements (including tasks, gateways, roles, and documents) and could benefit from decomposition into subprocesses. Breaking down the model into smaller, more manageable subprocesses would reduce complexity, improve understanding, and enhance adherence to this guideline.

Overall Summary:

- The overall adherence to the 7PMG guidelines suggests that while the process model has some strengths, there are several areas for improvement, particularly regarding simplicity, clarity, and structuredness. By reducing the number of elements, ensuring a single start and end point, improving the logical flow, and consistently using verb-object labels, the model could significantly increase its adherence to best practices.

Average Rating: 4.86/10

This indicates that the model needs substantial improvement to meet the 7PMG guidelines more effectively.

LLM-Output 6: Context-enhanced Assessment of Process Model Quality R3

R4

To analyze the correctness of the modeling in the attached process model, I will highlight the key elements that do not adhere to best-practice modeling standards, focusing on inconsistencies, redundancies, unclear connections, misused symbols, excessive complexity, and deviations from standard modeling conventions.

Observations and Analysis:

1. **Redundant Gateways and Decisions:**
   - There are multiple decision points or gateways asking similar questions, such as "Is a market analysis necessary?" and "Should any analysis be conducted or has that already happened?" These could potentially be combined or streamlined to reduce redundancy.
   - Questions like "Is the analysis complete?" and "Is the rough analysis complete?" seem to overlap in meaning and purpose, which might create confusion for readers and users of the model.
2. **Unclear Connections and Workflow Direction:**
   - Several connections between tasks do not clearly indicate the flow of the process. For instance, the transition between “Perform the rough analysis” and “Create risk analysis” does not make it evident whether these tasks are sequential or parallel.
   - The model uses ambiguous terms like "Start," "End," and "Project Kicked-Off" without a clear depiction of how these states interrelate or if they represent different levels or phases of the project.
3. **Misuse or Lack of Standard Symbols:**
   - The model uses generic shapes (rectangles) for different types of activities, decisions, and documents. BPMN standards typically suggest specific symbols: diamonds for decision points, rectangles with rounded corners for processes, and document icons for data objects.
   - There seems to be an inconsistency in symbol usage; for example, "Create risk analysis" and "Review process draft" are both processes but do not follow the same visual representation.
4. **Excessive Complexity:**
   - The process model appears cluttered with overlapping elements, such as tasks, data objects, and decision points, making it difficult to interpret quickly. For example, the inclusion of many "Start" and "End" labels creates visual noise and complicates understanding the overall workflow.
   - Several swimlanes (representing different roles or departments) appear overpopulated, making it challenging to track the specific responsibilities and workflow between different entities.
5. **Inconsistencies in Roles and Responsibilities:**

  - The model lists roles like "Project Manager," "Coordinator," "Quality Management," etc., but their responsibilities overlap or repeat in multiple tasks. For instance, "Project Manager" is repeatedly mentioned for various activities, which could be consolidated for clarity.
  - The term "Project Manager" is inconsistently capitalized throughout the document, which, while minor, suggests a lack of attention to detail in standardization.

6. **Ambiguity in Task Definitions and Scope:**

   - Tasks like "Perform a market analysis" and "Conduct a rough analysis for the project?" are vague and do not specify what constitutes completion or what criteria need to be met to proceed.
   - The model lacks clear definitions or criteria for decision-making steps, such as what constitutes a "necessary" analysis or a "detailed" plan.

7. **Deviations from Standard Modeling Conventions:**

   - There is a mix of task types (e.g., meetings, document creation, decision-making) without clear differentiation or categorization.
   - The use of color, shapes, or lines does not consistently follow standard conventions, making it hard to distinguish between different types of activities, decisions, or flows.

Recommendations for Improvement:

1. **Consolidate Redundant Elements:**

   - Merge similar decision points to streamline the decision-making process. For example, instead of multiple gateways asking about different analyses, have a single, multi-option decision point.

2. **Clarify Connections and Use Standard Symbols:**

   - Use arrows to clearly indicate the direction of the process flow. Adopt standard BPMN symbols (e.g., diamonds for decision points, rectangles for tasks) to differentiate between task types, events, and gateways.

3. **Simplify and Organize Swimlanes:**

   - Group related tasks under fewer swimlanes and ensure each lane represents a distinct role or department without excessive overlap.

4. **Define Clear Task Criteria and Scope:**

   - Provide clear descriptions for each task or decision point. For example, specify what the "rough analysis" entails or what determines if a "detailed plan" is necessary.

5. **Standardize Roles and Responsibilities:**

   - Use consistent naming conventions for roles throughout the model and define clear responsibilities for each role to avoid confusion.

6. **Reduce Excessive Complexity:**

   - Eliminate redundant "Start" and "End" points and focus on a single entry and exit point for each process to simplify the model.

By addressing these points, the process model will be more aligned with best practices, making it clearer, more understandable, and easier to maintain.

LLM-Output 7: Assessment of Process Model Quality R4

The Seven Process Modeling Guidelines (7PMG) provide a framework for evaluating the quality of process models. Let's evaluate the provided process model according to each of the guidelines, using the scale from 1 to 10, where 1 indicates poor adherence and 10 indicates excellent adherence.

Evaluation Based on 7PMG:

1. **G1: Use as few elements in the model as possible (Simplicity)**
   - **Rating: 4/10**
   - **Reason**: The model contains a high number of tasks, decision points, and data objects, leading to excessive complexity. There are redundant decision points and tasks that could be consolidated to reduce the number of elements. This complexity makes the model harder to understand and follow.
2. **G2: Minimize the routing paths per element (Minimality)**
   - **Rating: 5/10**
   - **Reason**: Many elements, especially decision points, have multiple outgoing and incoming connections. Some decisions lead to other decision points, creating a web of routes that could be minimized. The multiple paths per element make the model more difficult to navigate and understand.
3. **G3: Use one start and one end event (Structuredness)**
   - **Rating: 3/10**
   - **Reason**: The model uses multiple "Start" and "End" points, which creates confusion about the beginning and the end of the process. A good practice is to have a single start and end event to clearly define the process's entry and exit points. This model does not adhere to this guideline well.
4. **G4: Model as few parallel paths as possible (Cohesion)**
   - **Rating: 5/10**
   - **Reason**: The model includes some parallel paths, which could be reduced or represented differently to avoid confusion. The model lacks a clear indication of which tasks are concurrent, and it is not easy to determine the relationship between some parallel tasks. This can lead to misunderstandings about the process flow.
5. **G5: Avoid OR gateways where possible (Gateway Appropriateness)**
   - **Rating: 6/10**
   - **Reason**: While there are several decision points, most appear to be binary (yes/no) decisions, which is appropriate. However, there are instances where multiple decision points could be combined or simplified to avoid overusing gateways that complicate the model unnecessarily.
6. **G6: Use verb-object activity labels (Labeling Clarity)**
   - **Rating: 7/10**
   - **Reason**: Most activities in the model use clear verb-object labels (e.g., "Perform market analysis," "Create risk analysis"), which aligns with the guideline for clarity. However, some labels are vague or could be more specific to improve understanding (e.g., "Conduct rough analysis for the project?").
7. **G7: Decompose a model with more than 50 elements (Decomposability)**
   - **Rating: 3/10**
   - **Reason**: The model appears to contain well over 50 elements without decomposition into sub-processes or modular parts. This lack of decomposition results in a cluttered and complex model that is hard to interpret. Breaking it down into smaller, more manageable sub-models would significantly enhance clarity and maintainability.

Summary of Ratings:

- **G1: Simplicity** - 4/10
- **G2: Minimality** - 5/10
- **G3: Structuredness** - 3/10

- **G4: Cohesion** - 5/10
- **G5: Gateway Appropriateness** - 6/10
- **G6: Labeling Clarity** - 7/10
- **G7: Decomposability** - 3/10

Overall Assessment:

The process model shows moderate to poor adherence to most of the 7PMG guidelines, particularly in terms of simplicity, structuredness, and decomposability. Improvements are needed to reduce the number of elements, consolidate decision points, limit the use of multiple start/end events, and break the model into smaller sub-processes. However, it does perform relatively well in terms of labeling clarity and avoiding excessive use of OR gateways.
To align more closely with the 7PMG guidelines, a rework focusing on simplifying the process flow, reducing redundant elements, and enhancing clarity through structured decomposition is recommended.

LLM-Output 8: Context-enhanced Assessment of Process Model Quality R4

R5

Based on the provided process model from the PDF, I have identified several areas where the model does not adhere to best-practice modeling standards:

1. **Inconsistent Use of Gateways**
   - The model contains several decision points (e.g., "Is a market analysis necessary?", "Detailed analysis necessary?") that are marked with plain text rather than standardized decision symbols such as diamonds. This inconsistency can confuse readers and hinder understanding.
   - Some gateways have overlapping or redundant purposes, like having separate decisions for "Detailed analysis necessary?" and "Is the analysis complete?". This redundancy could be consolidated into fewer gateways to improve clarity.
2. **Redundant Elements**
   - Multiple instances of similar tasks appear without clear distinction or justification. For example, there are several points where "Review Process draft" or "Hold a presentation to Stakeholders" appear multiple times, creating redundancy without specifying if these are different iterations or stages of the same process.
   - The duplication of roles, such as the "Project Manager" appearing in nearly every task, can make it seem like one individual is responsible for all actions, which may not reflect reality or best practices in delegation and responsibility distribution.
3. **Unclear Connections and Flows**
   - Several connections between elements are not clearly defined, with lines crossing each other without explicit directional arrows or clear indications of the process flow. This makes it difficult to follow the progression from one step to the next.
   - The flow from decisions like "Changes necessary?" or "PR necessary?" does not clearly indicate what happens next if the answer is "No." This lack of clarity can lead to confusion about the process's outcome.
4. **Misuse of Symbols**
   - Some steps, such as "Start" and "End," are presented with the same symbols as intermediate processes, rather than using standard start/end symbols (typically circles or rounded rectangles in most process modeling standards).
   - Data elements (e.g., "Specification Sheet," "Risk Analysis") are not clearly differentiated from activities or tasks. They should be represented with symbols like documents or data stores to visually distinguish them from processes.

5. **Excessive Complexity**
   - The model appears to have too many elements on a single level, making it visually complex and potentially overwhelming for the reader. Consider breaking down the process into sub-processes or layers to simplify the visual representation.
   - The inclusion of too many roles without clear differentiation can create confusion. For example, the "Project Manager," "Coordinator," "IS Management," and others are mentioned without an apparent structure or hierarchy, leading to a lack of clarity regarding who oversees which tasks.
6. **Deviations from Standard Modeling Conventions**
   - Standard process modeling conventions, such as Business Process Model and Notation (BPMN), Unified Modeling Language (UML), or flowcharts, usually have strict guidelines on symbols and flow representations. This model deviates from those by using text descriptions rather than graphical symbols for key decisions and lacks a clear legend to explain the chosen symbols.

Recommendations for Improvement:

1. **Standardize Symbols:** Use standardized symbols for decisions, start/end points, data, and tasks to enhance readability and ensure the model adheres to BPMN or other standard modeling conventions.
2. **Simplify the Flow:** Reduce complexity by combining similar tasks, removing redundant elements, and breaking down the model into sub-processes where necessary.
3. **Clarify Connections:** Ensure all connections between elements are clear and directional, with explicit arrows indicating the flow of the process.
4. **Use Clear Labels:** Ensure all elements are clearly labeled, and consider including a legend to help users understand the symbols and terminology used.
5. **Consolidate Redundant Gateways:** Remove or merge gateways that serve similar purposes to streamline the decision-making points in the process.

By addressing these issues, the process model can become clearer, more concise, and more effective in communicating the intended process flow.

LLM-Output 9: Assessment of Process Model Quality R5

To evaluate the provided process model based on the Seven Process Modeling Guidelines (7PMG), I'll provide a rating for each guideline on a scale of 1 to 10, with 1 indicating poor adherence and 10 indicating excellent adherence.

1. **Guideline 1: Use as Few Elements in the Model as Possible (Simplicity)**
   - **Rating: 4/10**
   - The process model contains a large number of elements, some of which are redundant or unnecessary. There are multiple instances of similar tasks, decision points that could be combined, and a general lack of consolidation. This complexity reduces the model's clarity and makes it harder to understand at a glance.
2. **Guideline 2: Minimize the Routing Paths per Element (Structuredness)**
   - **Rating: 5/10**
   - The model has multiple routing paths that cross over each other and lack clear direction, which complicates the understanding of the workflow. Some decision points lead to ambiguous or unclear next steps, which violates the principle of minimizing complex routing paths. Improving the flow structure could help achieve a higher adherence score.
3. **Guideline 3: Use One Start and One End Event (Single Entry/Exit Point)**
   - **Rating: 3/10**

- There are multiple "Start" and "End" points in the model, which can confuse the reader about where the process begins and ends. A good process model should have a clear single entry and exit point to maintain clarity and avoid ambiguity.

4. **Guideline 4: Model as Structured as Possible (Structuredness)**

    - **Rating: 4/10**
    - The model lacks a structured approach in terms of hierarchical organization. Tasks, decisions, and subprocesses are not clearly organized or grouped logically. The presence of many different roles and responsibilities without clear grouping or flow structure detracts from a coherent and organized process model.

5. **Guideline 5: Avoid OR-Gateways (Use XOR and AND Instead)**

    - **Rating: 6/10**
    - There is some adherence to this guideline, as the process does not seem to contain explicit OR-gateways. However, the decision points and flow paths are ambiguous and do not consistently use XOR (exclusive OR) or AND (parallel split/join) gateways according to standard BPMN practices. This lack of clarity makes it challenging to determine if the OR-gateways are genuinely avoided.

6. **Guideline 6: Use as Few Elements in the Model as Possible on the Same Level (Decomposition)**

    - **Rating: 3/10**
    - The model presents too many elements on the same level, making it visually overwhelming and difficult to follow. The lack of decomposition into subprocesses or smaller, more manageable units results in a crowded and complex model. Breaking down the process into smaller sub-processes could significantly improve adherence to this guideline.

7. **Guideline 7: Use Naming Conventions Clearly (Naming Clarity)**

    - **Rating: 5/10**
    - The model uses generic names for tasks (e.g., "Hold a presentation" or "Create risk analysis") that do not provide enough context or detail. While some names are clear, others lack specificity and do not follow consistent naming conventions that would improve the model's readability and understanding.

Overall Assessment

- The model generally exhibits poor to average adherence to the Seven Process Modeling Guidelines. Key areas of improvement include reducing the number of elements, structuring the process more clearly, minimizing routing paths, using consistent start and end points, and adhering to standard naming conventions. A more focused application of these guidelines would greatly enhance the model's clarity, simplicity, and effectiveness.

Would you like to discuss specific areas of improvement in more detail or focus on any particular guideline?

LLM-Output 10: Context-enhanced Assessment of Process Model Quality R5

9.2 Appendix to Chapter 6

9.2.1 Process Models

Design

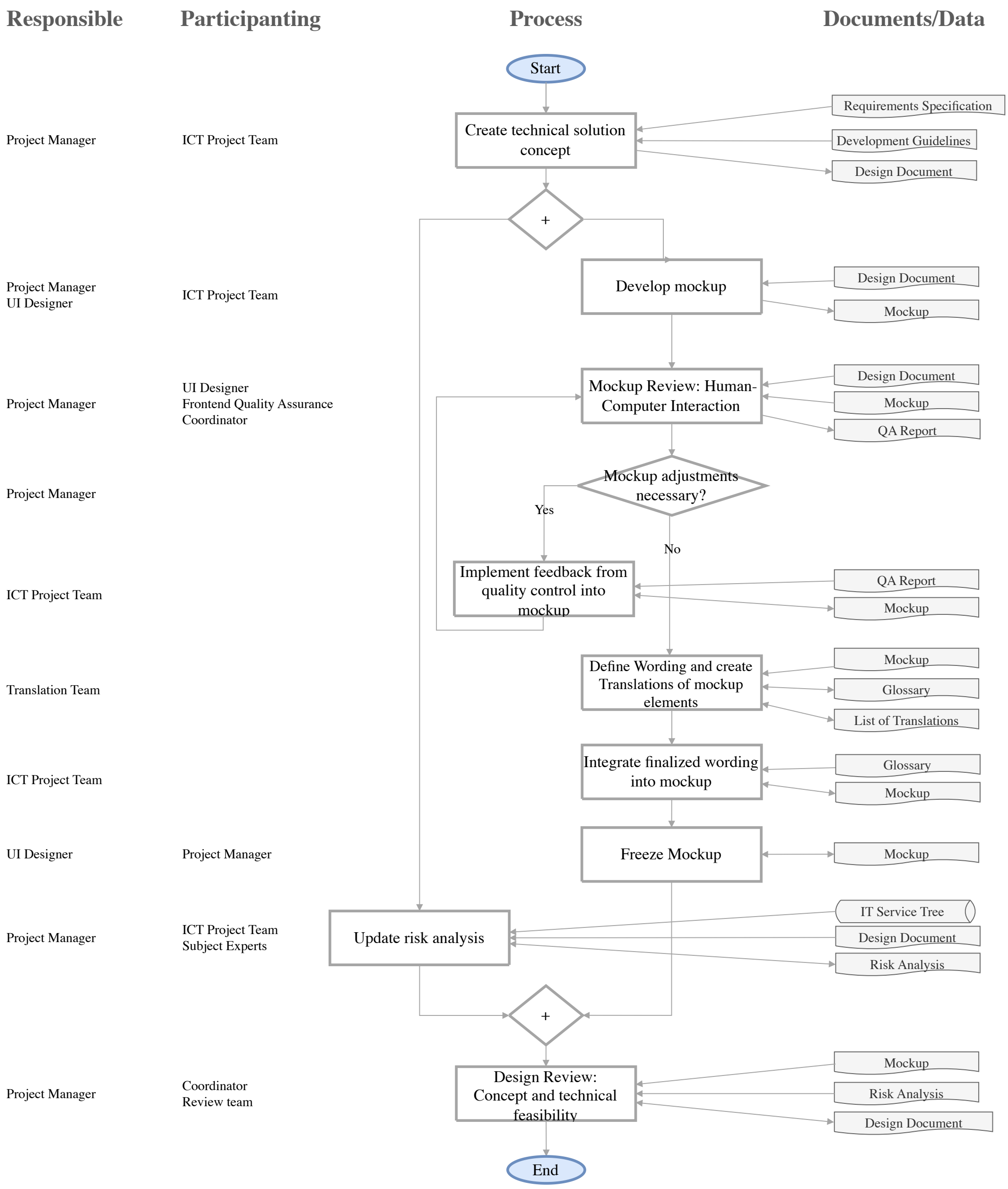

Figure 9.3: Original Design Process

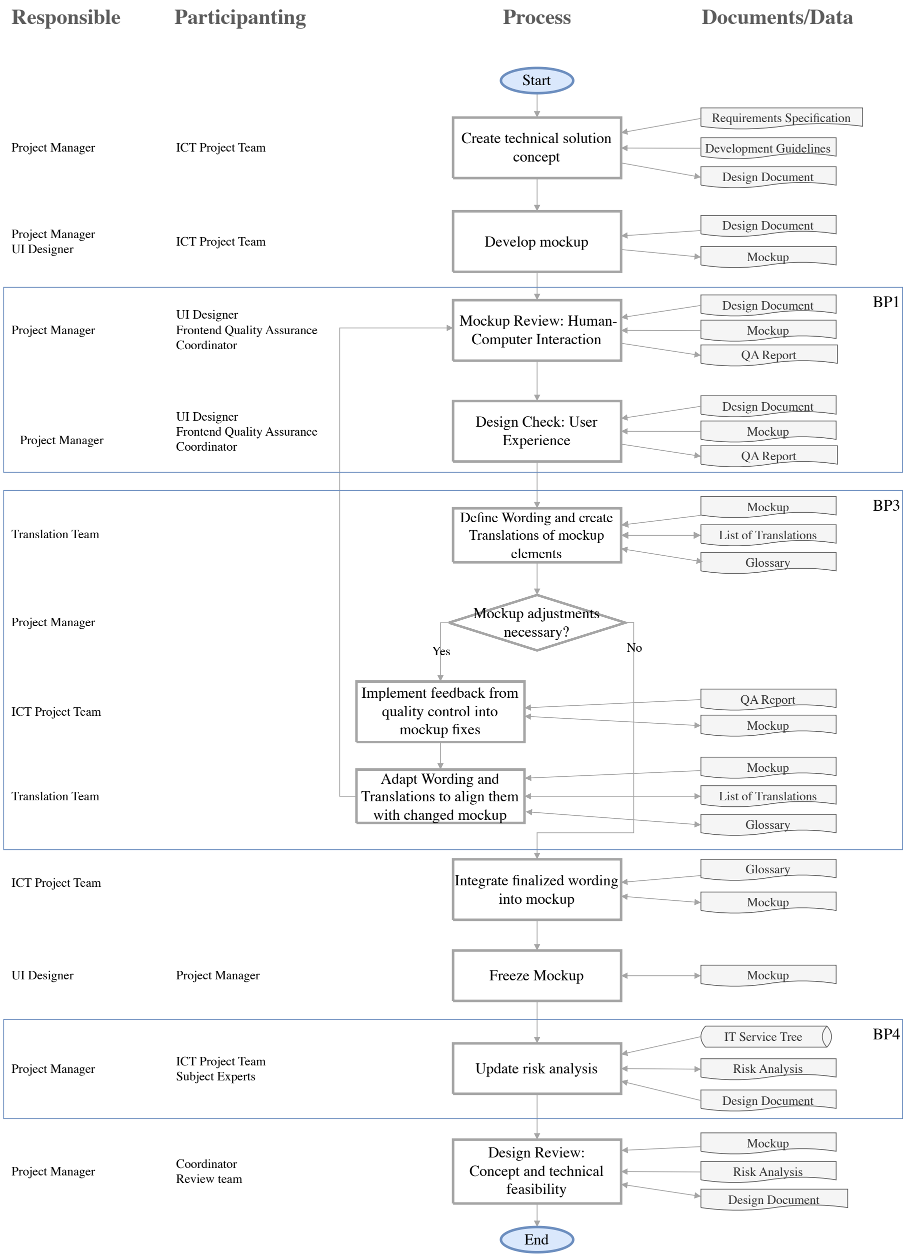

Figure 9.4: Adapted Design Process

Implementation

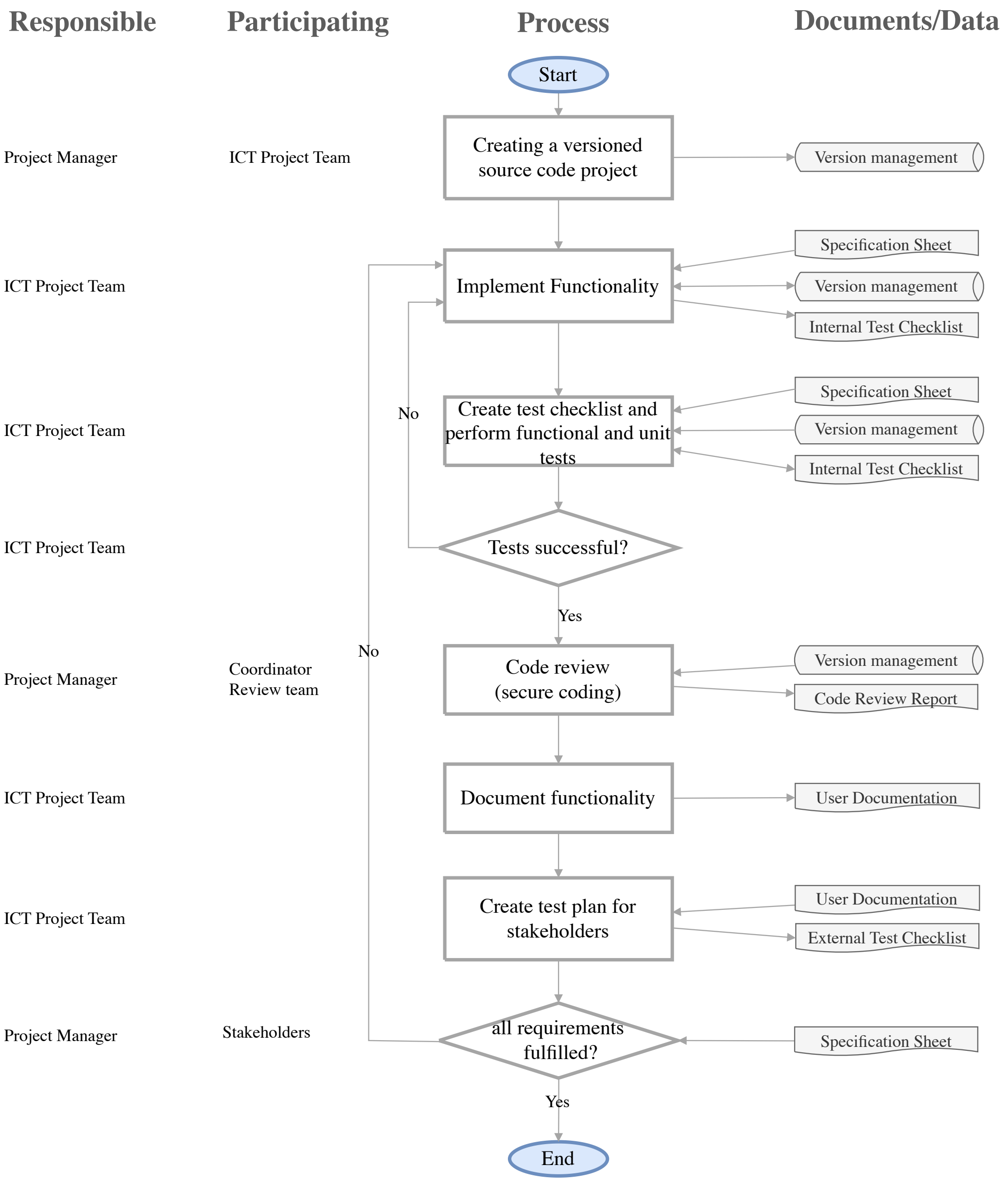

Figure 9.5: Original Implementation Process

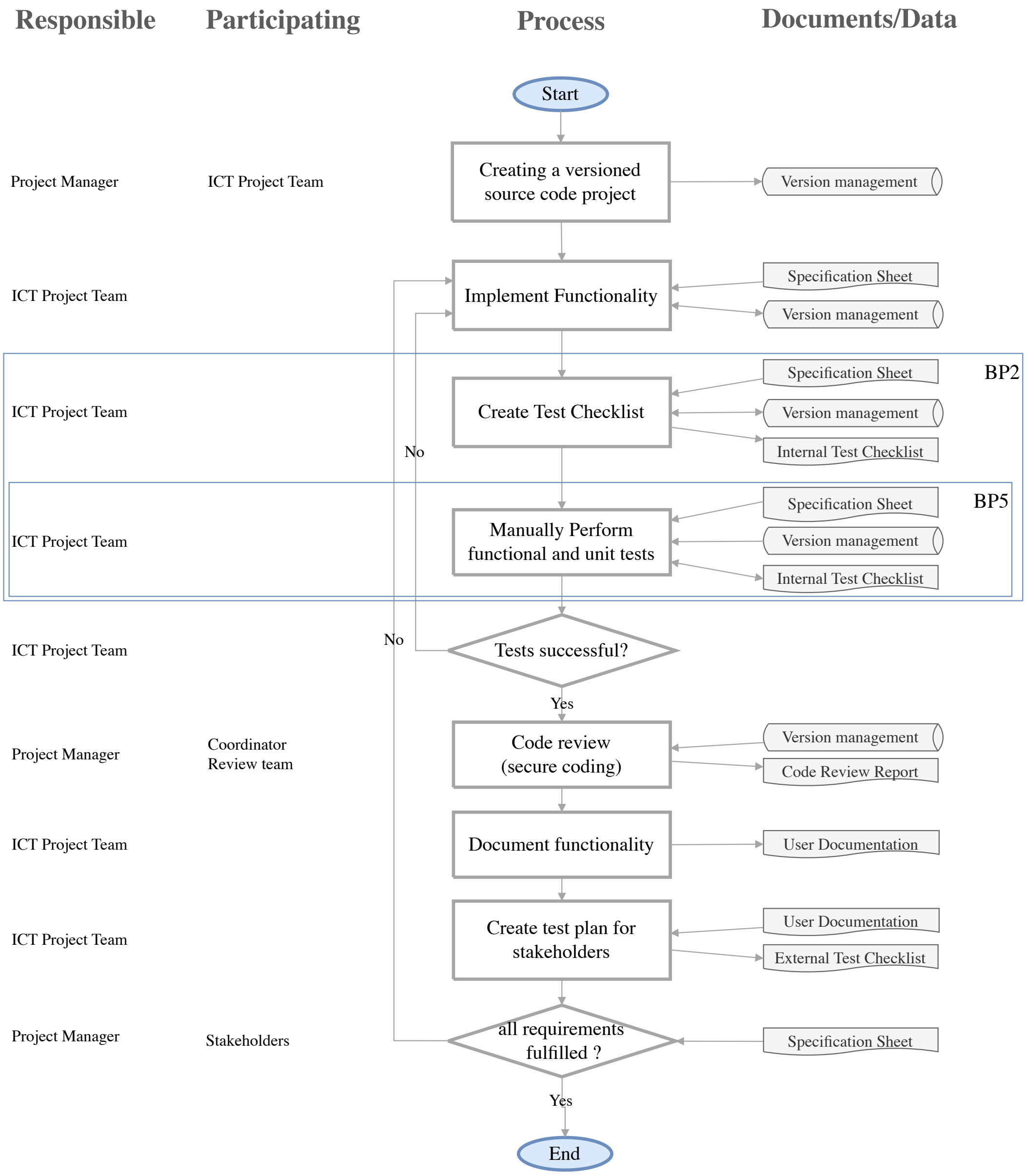

Figure 9.6: Adapted Implementation Process

Go-Live

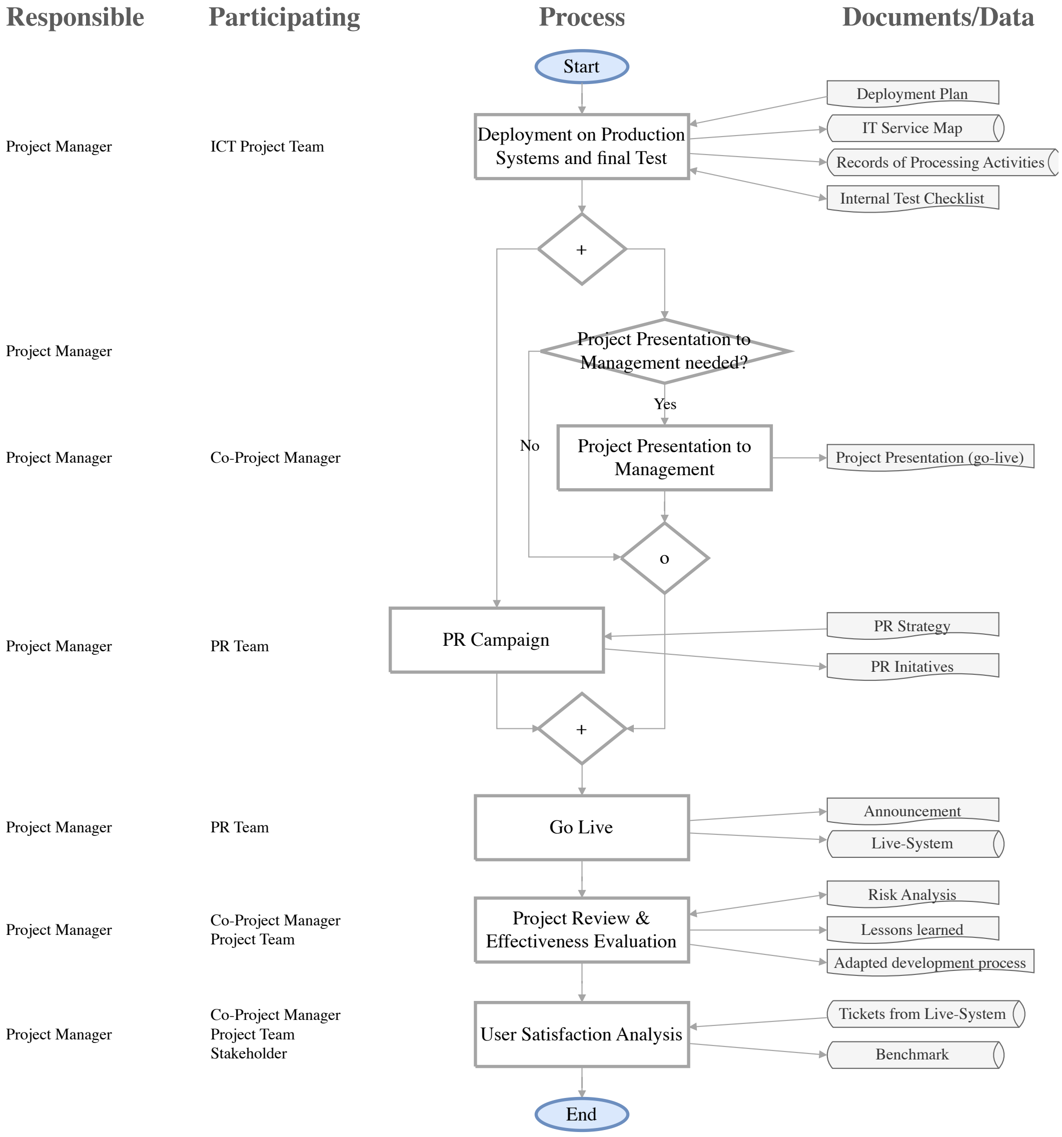

Figure 9.7: Original Go-Live Process

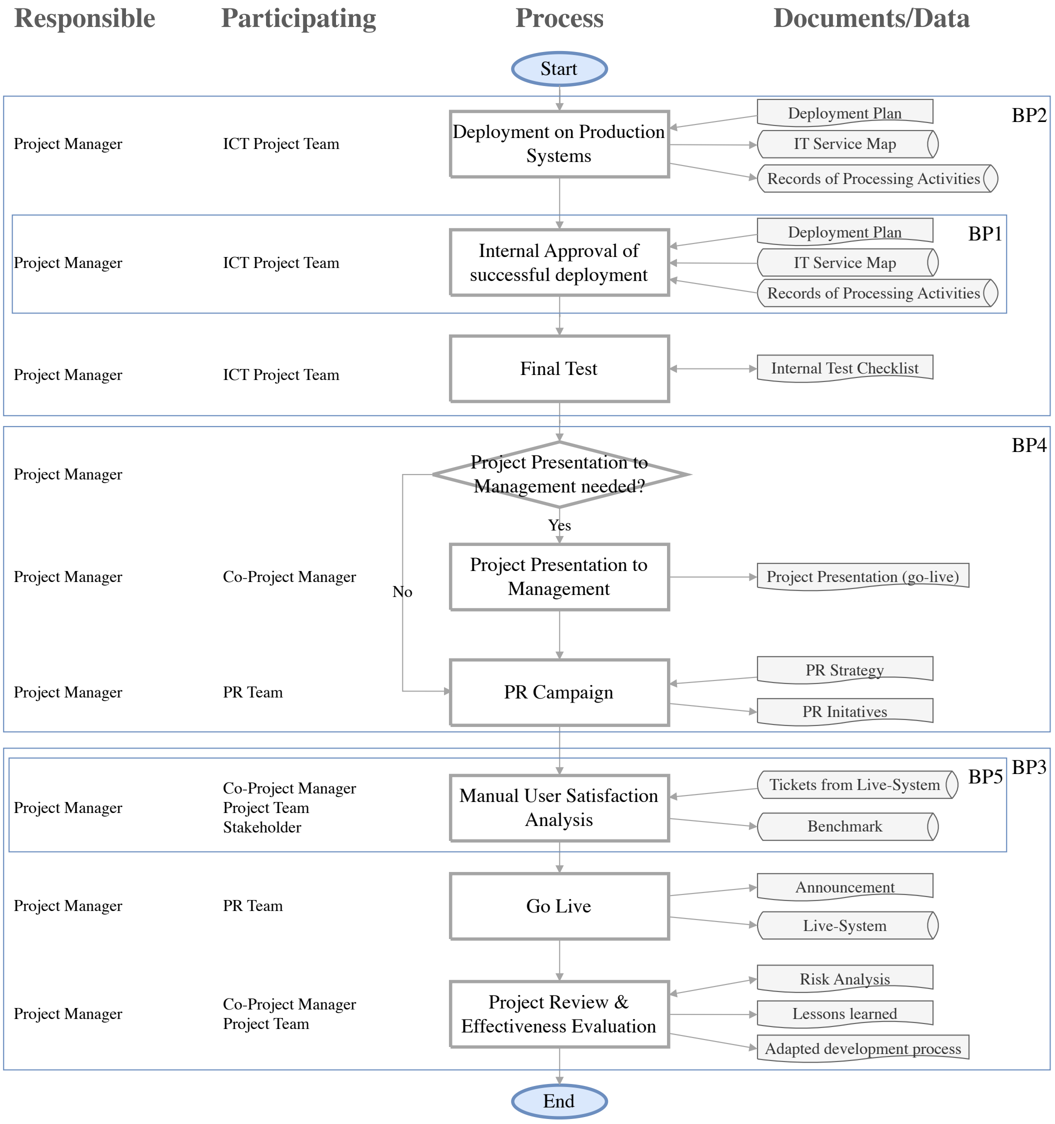

Figure 9.8: Adapted Go-Live Process

9.2.2 Model Outputs

PDF

Design Process

R1

To optimize the process outlined in the PDF, I will streamline certain steps while maintaining clarity and efficiency. Here’s what I would change:

Original Process

1. **Mockup adjustments necessary?**
2. **Integrate finalized wording into the mockup.**
3. **Risk Analysis.**
4. **Design Document: Concept and technical feasibility.**
5. **Update risk analysis.**
6. **Freeze Mockup.**
7. **Adapt Wording and Translations to align with the changed mockup.**
8. **Implement feedback from quality control into mockup fixes.**
9. **Mockup Review: Human-Computer Interaction.**
10. **Design Check: User Experience.**
11. **Create technical solution concept.**
12. **Develop mockup.**
13. **Define Wording and create Translations of mockup elements.**

Optimized Process

1. **Initial Mockup Adjustment Decision**
   Merged "Mockup adjustments necessary?” and "Integrate finalized wording into the mockup.”
   Reason: These steps logically follow one another, so they can be combined to avoid unnecessary handoffs.

2. **Risk Analysis Integration**
   Merged "Risk Analysis” and "Update risk analysis.”
   Reason: Updating the risk analysis should be part of the initial review, not a separate step, ensuring risks are identified and addressed upfront.

3. **Streamlined Mockup and Translation Update**
   Moved "Adapt Wording and Translations” right after "Freeze Mockup.”
   Reason: Reducing back-and-forth between teams by freezing the mockup before translation adjustments, preventing rework.

4. **Combine QA and Feedback**
   Merged "Implement feedback from quality control into mockup fixes” with "Mockup Review: Human-Computer Interaction” and "Design Check: User Experience.”
   Reason: These checks can be conducted simultaneously during a single review process to reduce time and improve coordination between QA and UX teams.

5. **Technical Concept and Mockup Development Together**
   Merged "Create technical solution concept” with "Develop mockup.”
   Reason: Developing the technical solution and mockup together ensures alignment between design and functionality.

6. **Translation Team Involvement Earlier**
   Moved "Define Wording and create Translations of mockup elements' earlier in the process (right after Mockup adjustments).
   Reason: This allows translation to be better integrated and reduces delays later on when translations may affect design elements.

Optimized Process Summary

1. **Mockup Adjustment and Wording Finalization.**
2. **Risk Analysis and Update.**
3. **Freeze Mockup and Adapt Translations.**
4. **Single Review: QA, UX, HCI.**
5. **Develop Technical Concept and Mockup.**
6. **Integrate Translations into Final Design.**

These optimizations streamline the process, reduce unnecessary handoffs, and encourage earlier collaboration between design and translation teams. This reduces duplication of efforts and improves overall project efficiency.

LLM-Output 11: Design Process Optimization (PDF) R1

R2

I will now optimize the process described in the document and outline the changes for you.

Original Process Steps

1. **Mockup adjustments necessary?**: Decision-making stage to assess if mockup adjustments are required.
2. **Integrate finalized wording into mockup**: Based on the mockup adjustments, the finalized wording is added.
3. **Risk Analysis**: Conduct a risk analysis for the mockup and design document.
4. **Design Review: Concept and technical feasibility**: Review the concept and assess technical feasibility.
5. **Freeze Mockup**: Finalize the mockup once all adjustments are incorporated.
6. **Mockup Review: Human-Computer Interaction**: Review the mockup for usability and human interaction.
7. **Create technical solution concept**: Generate a technical concept based on the mockup.
8. **Design Check: User Experience**: Conduct a user experience check.
9. **Wording and Translations**: Align translations with the final mockup.
10. **Implement feedback from QA**: Fix issues based on QA feedback.
11. **Finalize Design Document**: Produce the final design document.
12. **Complete Process**: End process.

Optimized Process

1. **Preliminary Review**: Before deciding if mockup adjustments are needed, introduce a pre-check phase that includes both technical feasibility and risk analysis. This consolidates initial steps and ensures that fewer iterations are required if adjustments are necessary.
   Change: Merged risk analysis with the first review of mockup adjustments.

2. **Parallel Tasks**: While adjusting the mockup, begin work on translating wording and aligning it with the new changes. This can be done in parallel rather than waiting for mockup approval to initiate translations.
   Change: Overlapping "Integrate finalized wording” and "Define Wording and create Translations.”

3. **Simplified Review Process**: Consolidate the Design Review (concept & technical feasibility) and the Human-Computer Interaction Review into a single, multi-disciplinary review. This reduces the number of separate reviews.
   Change: Combine "Design Review” and "Mockup Review” into one integrated review.

4. **Freeze Mockup Post-Review**: Ensure the mockup is frozen only after the combined review, instead of freezing it prematurely and needing subsequent changes.
   Change: Move the "Freeze Mockup” step to after the comprehensive review.

5. **Continuous QA Feedback Loop**: Implement continuous feedback from QA throughout the design process, rather than waiting for the end. This minimizes larger reworks at later stages.
   Change: Introduce earlier QA engagement instead of implementing all feedback after the mockup is complete.

6. **Finalize Process in One Step**: Remove the need for two separate steps for the design document. Once the design and mockup are finalized, the documentation can be completed and frozen in one step.
   Change: Combine "Finalize Design Document” and "Complete Process.”

Key Changes

- **Consolidation of risk analysis** and design review steps to streamline the workflow and reduce feedback loops.

- **Parallel processing of translations** during mockup adjustment instead of after mockup finalization.

- **Earlier involvement of QA** to ensure a continuous improvement loop, leading to less rework.

- **Combining steps** related to review processes, reducing the number of separate reviews and consolidating them for efficiency.

These optimizations aim to reduce redundant steps, enhance parallel processing, and incorporate feedback loops earlier to improve overall efficiency.

LLM-Output 12: Design Process Optimization (PDF) R2

R3

To optimize the process presented in the digitization workflow, I will focus on minimizing redundancy, improving communication, and streamlining decision-making steps. Here’s what I changed:

Changes Made

1. **Merge Redundant Reviews**:
   The process contains multiple reviews (e.g., "Mockup Review: Human-Computer Interaction" and "Design Check: User Experience"). Combining these reviews into a single comprehensive review could save time and effort, as these areas often overlap.
   Before: Separate reviews for Human-Computer Interaction and User Experience.
   After: Combined these reviews into a single "Design and User Experience Review.”

2. **Early Integration of Risk Analysis**:
   Instead of performing risk analysis after the design document is created, I suggest moving the "Update risk analysis” step to just after the mockup adjustments are made. This ensures that any changes are reflected earlier in the process.
   Before: Risk analysis occurs after mockup adjustments and document freeze.
   After: Risk analysis is done immediately after mockup adjustments, making risks clearer earlier in the process.

3. **Consolidate Feedback Loops**:
   Instead of having feedback implemented at multiple points, I recommend consolidating feedback into one phase after a major review, instead of spreading it throughout the workflow.

Before: Feedback implemented at multiple stages (Quality Control, Design Review).
After: A single feedback phase post "Design and User Experience Review" to address all issues at once.

4. **Parallel Tasks for Translations**:
   To avoid bottlenecks, I would recommend parallelizing the "Adapt Wording and Translations" with the "Mockup Adjustments" phase, instead of doing translations only after mockup freezing.
   Before: Translations after mockup adjustments.
   After: Translations and wording adaptation happening in parallel with mockup adjustments.

These changes aim to reduce duplication of effort, tighten the feedback loops, and make risk management more proactive while parallelizing tasks for greater efficiency.

LLM-Output 13: Design Process Optimization (PDF) R3

R4

To optimize the provided design process, I would suggest the following changes:

1. **Merge Redundant Steps**
   Risk Analysis appears multiple times in the process. Consolidating them into a single step can streamline the workflow and reduce duplications. Instead of repeating, use a single risk analysis step that applies to both the concept and technical feasibility stages.
   Change: Combine the separate instances of "Risk Analysis" into one common step.

2. **Parallel Execution**
   Mockup adjustments, **feedback implementation**, and **translation updates** can happen simultaneously instead of sequentially. This saves time by allowing multiple teams to work in parallel rather than waiting for one step to finish before starting the next.
   Change: Allow mockup fixes, translation updates, and quality control feedback implementation to be done in parallel.

3. **Feedback Loops**
   Currently, the feedback from the quality control team (QA) goes directly to mockup fixes without any intermediary step for approval or review. Adding a **review checkpoint** after implementing the fixes can ensure all feedback is appropriately integrated and validated.
   Change: Insert a "Mockup Fix Review" after implementing quality control feedback to ensure alignment with the overall design goals.

4. **Glossary and Translation Synergy**
   Rather than having the **glossary and translation** processes being isolated, there should be a step to synchronize them early on to avoid mismatches later in the process.
   Change: Move the translation process earlier to align better with the glossary's creation, ensuring the terminology used is consistent across the design process.

5. **Remove Overlapping Roles**
   The **Project Manager** is listed in multiple areas with overlapping responsibilities. Delegating specific tasks to other specialized roles (e.g., translation team or QA coordinator) can free up the project manager's capacity to focus on overall coordination rather than micromanaging.
   Change: Delegate specific responsibilities to team members, removing the Project Manager from redundant roles.

Final Optimized Process

1. Start with **Mockup Review**.

2. **Consolidate Risk Analysis** into one step.

3. Proceed with **Design Document and Concept Feasibility Check**.

4. **Allow parallel work** on feedback integration, mockup fixes, and translation updates.

5. Add a **checkpoint for reviewing mockup fixes** after QA feedback is applied.

6. Ensure **glossary and translation synchronization** early in the process.

This revised process reduces redundancies, allows more parallel work, and improves overall efficiency.

LLM-Output 14: Design Process Optimization (PDF) R4

R5

Here’s an optimization of the process from the document you’ve provided:

Key Changes

1. **Combined Steps:**
   The steps of "Define Wording and create Translations” and "Adapt Wording and Translations to align with changed mockup” are combined into one continuous task to avoid unnecessary back-and-forth and save time.

2. **Parallelizing Tasks:**
   Tasks like "Develop mockup” and "Risk Analysis” can be done in parallel. This ensures that risk analysis and design review are not sequential but rather simultaneous, reducing delays.

3. **Risk Analysis Streamlined:**
   The document currently calls for a "Risk Analysis” twice. I streamlined this by introducing continuous risk assessment throughout the project, instead of having distinct checkpoints. This allows for early detection and mitigation of risks.

4. **Mockup Iterations:**
   "Mockup Review: Human-Computer Interaction” is consolidated with "Design Check: User Experience” to reduce redundant review cycles. This ensures that user interaction design and technical feasibility are both considered in the same iteration.

5. **Quality Control Feedback Integration:**
   The task "Implement feedback from quality control” is shifted to occur after "Mockup Freeze.” This ensures that final mockup adjustments are informed by QA feedback and minimize the need for post-finalization adjustments.

Optimized Process Flow

1. **Start**

2. **Create technical solution concept**
   (In parallel with continuous risk analysis)

3. **Develop mockup**

4. **Define wording and create translations (aligned with the mockup)**

5. **Design Check: Human-Computer Interaction & User Experience**

6. **Mockup adjustments (if necessary)**

7. **Mockup Freeze**

8. **Quality Control (QA) Feedback Implementation**

9. **End**

This version removes redundant steps, reduces wait time between processes, and makes better use of parallel tasks, which should save time and resources.

LLM-Output 15: Design Process Optimization (PDF) R5

Bitmap

Design Process

R1

To optimize the provided process, I would suggest the following adjustments, keeping in mind the goal of improving efficiency, reducing unnecessary steps, and improving the clarity of roles:

1. **Combining Review Stages**
   - **Current Process:** Separate steps for "Mockup Review: Human-Computer Interaction" and "Design Check: User Experience."
   - **Optimization:** Combine these into a single review stage, "Mockup Review: HCI and User Experience," to streamline the feedback process. This will reduce back-and-forth delays and ensure that both aspects are reviewed simultaneously.
   - **Impact:**Time savings, improved collaboration, and more cohesive design feedback.
2. **Parallel Work on Wording and Mockup Adjustments**
   - **Current Process:** Wording and translations are done after both HCI and UX reviews are complete.
   - **Optimization:** Start defining wording and translations simultaneously with the "Mockup Review: HCI and User Experience" step. If mockup adjustments are needed, only a portion of the wording/translation may need to be updated.
   - **Impact:** Reduced idle time for the translation team and shorter project timelines.
3. **Remove Redundancy in Translation Updates**
   - **Current Process:** Separate steps for "Define Wording and Translations" and "Adapt Wording and Translations."
   - **Optimization:** Combine these into one iterative step that handles translations continuously as the mockup evolves.
   - **Impact:** Eliminate duplicated efforts, save time, and ensure that translations are always aligned with the current mockup version.
4. **Freeze Mockup at Earlier Stage**
   - **Current Process:** The mockup is frozen after all wording and translation adjustments are done.
   - **Optimization:** Freeze the mockup right after the "Mockup Review: HCI and User Experience" stage, and only adapt wording/translation afterward. This prevents constant rework due to changing mockup designs.
   - **Impact:**Stabilizes the design earlier in the process, allowing translations and risk analysis to proceed without constant changes.
5. **Improve Collaboration and Communication**
   - **Current Process:** Each team (ICT Project Team, Translation Team, etc.) works sequentially on tasks.
   - **Optimization:** Introduce regular short feedback loops, such as daily or bi-weekly sync meetings, involving all key participants. This will ensure issues are identified and resolved earlier, and all teams are on the same page.
   - **Impact:** Reduces miscommunication and backtracking.
6. **Automate Risk Analysis with Mockup Freezing**
   - **Current Process:** Risk analysis is a separate step after freezing the mockup.
   - **Optimization:** Automate portions of risk analysis once the mockup is frozen, with a focus on integrating risk elements directly into the mockup development tool or system. This reduces the time spent in a manual process and ensures faster risk assessment.
   - **Impact:** Quicker turnaround for risk analysis and better alignment with the frozen mockup.

Revised Process Flow:

1. Start
2. Create technical solution concept
3. Develop mockup
4. **Mockup Review: HCI and User Experience** (combined)
5. **Parallel Step:** Define Wording and Translations (start early)
6. **Decision:** Mockup adjustments necessary?
    - Yes: Implement feedback and adjust wording/translation in one step
    - No: Proceed to next step
7. Freeze Mockup
8. Update Risk Analysis
9. Final Design Review
10. End

These changes aim to reduce duplication, ensure parallel work where feasible, and improve coordination between teams, ultimately shortening the overall project duration while maintaining quality.

LLM-Output 16: Design Process Optimization (Bitmap) R1

Given the additional information, the process optimization can be further refined to accommodate the high demand on the translation team and the need for evolving risk analysis. Here's a revised approach:

1. **Deprioritize Perfect Wording for Mockups**
    - **Current Process:**
        - Translations and wording are carefully defined for every mockup, even though perfect wording isn't necessary.
    - **Optimization:**
        - Only essential terms that impact Human-Computer Interaction (HCI) or functional clarity should be translated or refined. Skip non-essential translation work until the final stages. This minimizes the translation team's involvement and workload.
    - **Impact:**
        - Less strain on the translation team, freeing them for higher-priority tasks, while maintaining usability in early mockup stages.
2. **Integrate Continuous Risk Analysis**
    - **Current Process:**
        - Risk analysis happens after the mockup is frozen.
    - **Optimization:**
        - Embed a real-time, evolving risk analysis process directly into the mockup development phases. This could involve automated tools or ongoing risk evaluations by the project manager as the design progresses, especially focusing on risk elements related to HCI, usability, or critical functional issues.
    - **Impact:**
        - Management gets continuous risk insights, ensuring that risks are addressed earlier and with greater flexibility throughout the design process, avoiding delays and bottlenecks at the end.

3. **Further Reduce Translation Burden**
    - **Current Process:**
        - Full translations and wording reviews happen during the "Define Wording and Translations" step.
    - **Optimization:**
        - Only initiate translation tasks after the mockup design has reached a stable point where no further HCI or UX changes are expected. This approach delays non-essential translation work until it's truly necessary, significantly reducing the workload for the translation team.
    - **Impact:**
        - The translation team focuses on essential tasks, reducing their overall workload and avoiding unnecessary rework when mockups change frequently.
4. **Ensure Parallel Processes**
    - **Current Process:**
        - Sequential steps delay overlapping work.
    - **Optimization:**
        - Enable parallel workflows between mockup design and risk analysis. For example, while UI Designers are focused on HCI, the project manager and quality assurance teams can concurrently begin evaluating potential risks, feeding real-time feedback into the mockup process. This also allows for incremental translation work, but only if it's absolutely necessary for critical elements.
    - **Impact:**
        - Faster project turnaround and better alignment with management's expectation for a continuously evolving risk analysis.

Revised Process Flow (with further adjustments):

1. Start
2. Create technical solution concept
3. Develop mockup
4. **Mockup Review: HCI (Focus on HCI only)** (User Experience can be delayed if necessary)
5. **Parallel Step:** Define essential wording and translations only for HCI-critical elements
6. **Continuous Step:** Risk analysis ongoing during HCI review and early mockup development
7. Decision: Are major adjustments necessary?
    - Yes: Implement feedback and update essential translations if critical to HCI
    - No: Proceed
8. Freeze Mockup
9. Finalize translations and wording for key elements only
10. Update final risk analysis (if necessary)
11. Final Design Review: Concept, HCI, and technical feasibility
12. End

Key Benefits:

- **Efficient Translation Efforts:** Minimized workload for the translation team, focusing only on crucial elements, reducing bottlenecks and rework.
- **Continuous Risk Analysis:** Embedded throughout the mockup process to align with management's expectations, avoiding delays at the end.
- **Streamlined Review Process:** Emphasis on HCI in earlier phases with less focus on perfect wording or UX initially, allowing iterative refinement without overwhelming teams.

This approach ensures that both translation efforts and risk analysis evolve dynamically and effectively throughout the process.

LLM-Output 17: Context-enhanced Design Process Optimization (Bitmap) R1

R2

To optimize the process presented in the image, several adjustments can be made for clarity, efficiency, and reducing potential bottlenecks. Here’s a breakdown of suggested changes:

Combine or Reorder Tasks to Avoid Duplication

- **Mockup Review (Human-Computer Interaction) and Design Check (User Experience)** are two sequential reviews that could be combined. Since both tasks involve reviewing the usability and interaction of the mockup, merging these into a single "Design Review: UX and HCI” process would streamline review phases and reduce the need for two separate reports.

  Change: Combine "Mockup Review: Human-Computer Interaction” and "Design Check: User Experience” into one task called **"Mockup Review: Usability and User Experience.”**

Clarify Feedback Loops

- In the current flow, after "Define Wording and Create Translations of Mockup Elements,” if adjustments are needed, the process loops back to "Implement Feedback.” However, there is no indication if the adjustments necessitate another translation or review cycle.

  Change: Add a specific feedback loop between "Adapt Wording and Translations” and "Mockup Review: Usability and User Experience” to ensure that any translations or wording changes are reviewed again before the final mockup freeze.

Parallel Processing to Save Time

- Instead of sequentially working on wording, translations, and mockup adjustments, tasks such as **Wording Definitions** and **Translation of Mockup Elements** could be carried out in parallel to the "Implement Feedback from Quality Control” phase.

  Change: Make the "Define Wording and Create Translations” task parallel to "Implement Feedback from Quality Control,” so the two teams can work simultaneously, reducing idle time.

Task Role Optimization

- Some tasks have overlapping responsibilities or could be simplified by better role distribution. For instance, **translation tasks** could be directly integrated into the review steps rather than creating a separate process flow for translations after each review.

  Change: Assign the Translation Team to collaborate with the UI Designer during the initial mockup phase, so translations are built into the review process early on. This would prevent having to return later to align translations with changes.

Risk Analysis and Mockup Freeze

- The "Freeze Mockup” happens before updating the risk analysis, which is problematic since potential risks could arise that would require further design adjustments.

  Change: Swap the steps for **"Freeze Mockup”** and **"Update Risk Analysis”** to ensure all risks are addressed before freezing the mockup design.

Final Review Optimization

- The current "Design Review: Concept and Technical Feasibility” includes many stakeholders, which could result in a drawn-out approval process. It might be more efficient to implement a pre-review phase with the **UI Designer** and **Project Manager** before the full team review.

  Change: Add a "Pre-Review” phase with the UI Designer and Project Manager to ensure that only finalized concepts reach the broader review team. This reduces redundant feedback and ensures only refined concepts are reviewed.

Summary of Changes

1. **Combine UX and HCI reviews** into one step.
2. **Create a feedback loop** for translations and wording changes after mockup adjustments.
3. **Parallel processing** for feedback implementation and translation steps.
4. **Integrate translations** earlier in the process with UI design.

5. **Switch "Freeze Mockup”** and **"Update Risk Analysis”** to address risks before freezing the design.

6. Add a **Pre-Review phase** to filter feedback before the full review team is involved.

These changes should streamline the process, reduce the number of unnecessary loops, and improve the overall efficiency of the design review cycle.

LLM-Output 18: Design Process Optimization (Bitmap) R2

Considering these additional factors, further optimizations can be made to the process to ensure alignment with the goals and constraints you've shared:

1. **Prioritize Human-Computer Interaction and Minimize Wording Perfection**
   - **Mockup Review: Usability and User Experience:** Since the mockups prioritize Human-Computer Interaction and do not need perfect wording, the emphasis should remain on the functional aspects and user experience during the review process. The translation team's involvement should be delayed until the mockup's structure and design are locked in, which reduces back-and-forth adjustments involving wording.
   - **Change:** Remove the need for **"Define Wording and Translations of Mockup Elements"** until after the mockup passes its final usability and design reviews. Translations should occur only after the design and interaction aspects are finalized.

2. **Evolve the Risk Analysis with the Design**
   - Since management requires real-time risk updates, the process should reflect that the **risk analysis evolves alongside the mockup development** rather than waiting until after the mockup is frozen. Integrating risk assessment earlier ensures potential design risks are identified and mitigated promptly, improving decision-making.
   - **Change:** Introduce **"Risk Analysis Updates"** as a continuous task alongside each review step (HCI/UX, Mockup Fixes) to ensure management receives real-time feedback. These updates should be made collaboratively with the **ICT Project Team** and **Subject Experts** to address technical risks that might emerge during development.

3. **Reduce the Translation Team's Involvement**
   - Given the high workload of the translation team, translations should occur only when absolutely necessary and after major design iterations are complete. Instead of having them adjust wording throughout the process, **translations can be a final task** carried out after the mockup is frozen and the risk analysis is complete.
   - **Change:** Move the **"Define Wording and Create Translations of Mockup Elements"** to the very end, just before the final **"Design Review,"** reducing the number of times the translation team is involved. This prevents them from having to revise translations multiple times due to design changes.

4. **Risk Assessment Integration**
   - Integrating risk analysis with the design phase helps management get real-time updates on emerging risks and ensures that changes made to the mockup are aligned with the risk profile of the project.
   - **Change:** Include **"Ongoing Risk Updates"** after every key decision or mockup change (e.g., "Mockup Review," "Design Check"), ensuring risks are continuously monitored. This eliminates the final step where risk analysis happens after the mockup freeze, as the analysis should already reflect all the finalized design decisions.

Final Optimized Process:

1. **Mockup Review: Usability and User Experience:** Prioritize HCI without worrying about perfect wording.

2. **Real-Time Risk Updates:** Introduce ongoing risk updates after each major review or mockup change, aligning with management's requirement for evolving risk analysis.

3. **Delayed Translations:** Involve the translation team only after the mockup is frozen and finalized to minimize unnecessary workload.

4. **Risk Mitigation:** Focus on technical and interaction-related risks during mockup development, integrating risk assessments into the design cycle rather than waiting for the end of the process.

This optimized process reduces the burden on the translation team, aligns with management's evolving risk analysis requirement, and keeps the focus on the core elements of HCI and UI design without letting wording perfection slow things down.

LLM-Output 19: Context-enhanced Design Process Optimization (Bitmap) R2

R3

Here is an optimized version of the process based on your provided diagram, with clear explanations of the changes made for efficiency and clarity:

Changes Made

1. **Parallel Tasks for Translation and Mockup Development**:

   - Moved "Define Wording and create Translations of mockup elements” earlier in the process, during the "Develop mockup” phase. This allows the translation team to work in parallel, reducing waiting times for both teams.

2. **Eliminated Redundant Review Step**:

   - The "Mockup Review: Human-Computer Interaction” and "Design Check: User Experience” phases were merged into a single, more comprehensive review step called "Design Review: UX and HCI.” This avoids duplication of effort and ensures a holistic review is conducted at once.

3. **Risk Analysis Moved to Earlier Phase**:

   - The "Update Risk Analysis” task is now performed earlier, right after the "Mockup adjustments necessary?” decision step. By updating risk early, potential problems can be addressed before the mockup is frozen, reducing the chances of having to revisit mockup design due to risk factors.

4. **Consolidated Feedback Loop**:

   - The "Implement feedback from quality control into mockup fixes” and "Adopt wording and translations to align them with changed mockup” steps were merged into a single "Implement Feedback and Finalize Mockup.” This prevents an unnecessary division of steps and allows faster integration of feedback and translation changes in one go.

5. **Simplified Document Updates**:

   - The final "Design Review: Concept and technical feasibility” now includes an automatic update of relevant documents (IT Service Tree, Risk Analysis, Design Document), instead of requiring separate updates for each document.

6. **Incorporation of Continuous QA**:

   - Quality Assurance (QA) has been integrated into the "Develop mockup” phase as an ongoing task, ensuring that feedback is continuously provided, preventing last-minute large changes after the mockup is nearly completed.

Optimized Process Flow

1. **Start**

2. **Create technical solution concept**

   - (Documents: Requirements Specification, Development Guidelines, Design Document)

3. **Develop mockup**
    - (Incorporate continuous QA feedback)
    - (Documents: Design Document, Mockup)
    - **Parallel**: Define wording and create translations of mockup elements
    - (Documents: Mockup, List of Translations, Glossary)
4. **Design Review: UX and HCI**
    - (Documents: QA Report, Mockup, Design Document)
5. **Mockup adjustments necessary?**
    - **Yes**: Proceed to feedback implementation
    - **No**: Proceed to finalization
6. **Implement Feedback and Finalize Mockup**
    - (Documents: QA Report, Mockup, List of Translations, Glossary)
7. **Update Risk Analysis**
    - (Documents: Risk Analysis, Mockup)
8. **Freeze Mockup**
    - (Documents: Mockup)
9. **Design Review: Concept and technical feasibility**
    - **Final step**: Documents updated automatically
    - (Documents: IT Service Tree, Risk Analysis, Design Document)

This streamlined process removes redundancies and allows tasks to be completed in parallel, resulting in a faster overall development cycle without sacrificing quality assurance or risk management.

LLM-Output 20: Design Process Optimization (Bitmap) R3

Given the additional considerations, here is a refined optimization that addresses both the focus on Human-Computer Interaction (HCI) and the need for evolving risk analysis, while also minimizing the workload of the translation team:
Adjustments to the Process:

1. **Focus on HCI Without Early Wording Finalization:**
    - Since the mockups prioritize HCI, perfect wording is not required upfront. The translation team should only be involved when the mockup design is near finalization to avoid unnecessary changes that require translation updates. This reduces the translation team's workload.
2. **Evolving Risk Analysis Throughout the Process:**
    - To align with management's requirement for a real-time evolving risk analysis, the risk assessment should be integrated into key decision points and updated incrementally as the design progresses. This ensures that risks are tracked continuously, without waiting for the final stages.
3. **Translation Team Engages Later:**
    - The translation team's involvement should be deferred until after the first round of feedback from the Design Review. This prevents them from translating elements that are likely to change, minimizing rework. Any minor wording required during HCI design reviews can be handled by the project team, as perfect wording is not necessary at that stage.

Optimized Process Flow (with adjustments):

1. **Start**
2. **Create technical solution concept**
   - (Documents: Requirements Specification, Development Guidelines, Design Document)
   - *Begin Initial Risk Analysis*
3. **Develop mockup**
   - (Incorporate continuous QA feedback with a focus on HCI)
   - (Documents: Design Document, Mockup)
   - *Ongoing Risk Analysis Updates*
4. **Design Review: UX and HCI**
   - *UI Designers and HCI Experts finalize interaction elements*
   - *Light feedback on wording as necessary by project team*
   - *Risk analysis is updated based on review results*
   - (Documents: QA Report, Mockup, Design Document, Risk Analysis)
5. **Mockup adjustments necessary?**
   - *Yes*: Implement feedback
   - *No*: Proceed to finalization
6. **Implement Feedback and Finalize Mockup**
   - Only when the mockup is stable:
   - **Define wording and create translations of mockup elements**
   - (Documents: Mockup, List of Translations, Glossary)
7. **Integrate finalized wording into mockup**
   - (Documents: Glossary, Mockup)
   - *Final Risk Analysis update*
8. **Freeze Mockup**
   - (Documents: Mockup, Risk Analysis)
9. **Design Review: Concept and technical feasibility**
   - *Final step*: Documents updated automatically
   - (Documents: IT Service Tree, Risk Analysis, Design Document)

Key Benefits of These Adjustments:

1. **Reduced Workload on Translation Team:** The translation team is only involved after the design is nearly finalized, preventing the need for frequent updates due to ongoing design changes. Minor wording for early-stage reviews can be managed by the project team, especially since perfect wording is not crucial at the HCI stage.

2. **Continuous Risk Management:** Risk analysis evolves alongside the design and is updated at each key review point, ensuring that management's expectations are met without waiting until the end of the process. This minimizes the risk of overlooking issues that could arise later.

3. **Streamlined Reviews with Focus on Interaction:** The review process is centered around the core priority: Human-Computer Interaction. Design and usability elements are finalized before bringing in other aspects such as translation, making the process more efficient.

By integrating risk updates throughout and strategically involving the translation team later in the process, this optimization maximizes efficiency while keeping the focus on HCI and minimizing unnecessary workload.

LLM-Output 21: Context-enhanced Design Process Optimization (Bitmap) R3

R4

To optimize this process, I would suggest focusing on the following key areas, with corresponding changes:

1. Parallel Task Execution

- **Current Issue:** Several tasks, especially regarding reviews and adjustments, occur in sequence, causing delays.
- **Change:** Parallelize the mockup development and translation tasks. For instance, the translation team can start creating translations as soon as the basic elements of the mockup are ready, instead of waiting for the entire design review to finish. This speeds up the workflow.

2. Feedback Loops

- **Current Issue:** The feedback loop occurs too late in the process (after translations and wordings are defined).
- **Change:** Include an earlier stage for feedback collection and adjustments after the first draft of the mockup. This minimizes the need for rework on translations and wording once the mockup is finalized.

3. Role Consolidation

- **Current Issue:** Several tasks involve both the Project Manager and UI Designer together, which could introduce inefficiency.
- **Change:** Assign clearer responsibilities to streamline decision-making. For example, delegate mockup adjustments based on quality assurance feedback solely to the UI Designer, with the Project Manager focusing on the technical and risk elements.

4. Translation Efficiency

- **Current Issue:** Wording and translations might need multiple revisions based on mockup adjustments.
- **Change:** Instead of waiting for the full list of translations, use agile iterations for translations, with partial reviews before major adjustments. This could involve integrating real-time tools for translation consistency, cutting down back-and-forth revisions.

5. Combine Risk Analysis and Freeze Steps

- **Current Issue:** Risk analysis and mockup freezing are separate stages.
- **Change:** Combine risk analysis with the freezing of the mockup. Freezing the mockup after addressing risk would ensure fewer iterations, reducing overall cycle time.

6. Tools & Automation

- **Current Issue:** The process heavily relies on manual quality checks and feedback.
- **Change:** Introduce automated testing tools for design consistency and user experience checks. This can be done after the "Develop Mockup" phase to reduce manual effort during the review stage.

Proposed Optimized Process Flow

1. **Create technical solution concept** → no change.
2. **Develop mockup** → use partial mockups to proceed with parallel translation.
3. **Parallelize Wording & Translation** → Start translation during early mockup design.
4. **Mockup Review & Feedback** → Feedback earlier in the process.
5. **Mockup Adjustment and Revision** → Role reallocation.
6. **Automate QA Checks** → Implement automated checks to avoid manual delays.
7. **Integrated Mockup with Translation** → Combine risk assessment and freezing stages.

These changes streamline the workflow, reduce manual steps, and allow for parallel tasks, which speeds up the entire process from concept to review.

LLM-Output 22: Design Process Optimization (Bitmap) R4

Given this additional information, I would suggest a refined optimization of the process to focus on the following priorities:

1. **Minimize Translation Team Workload**
    - **Current Issue:** The translation team is involved in creating translations for mockups, even though the designs do not require perfect wording.
    - **Change:** Reduce the translation team's role during early mockup stages. Only involve them at the **final stage** when the wording is essential and stable. Early mockups can use placeholder text to avoid unnecessary translation efforts. Only when mockups are ready for review or freeze should the translation team step in.
2. **Focus on Human-Computer Interaction (HCI) and UI Designers' Role**
    - **Current Issue:** The process heavily involves project managers in mockup reviews, which could slow down HCI-focused design work.
    - **Change:** Let UI Designers take full ownership of the mockup review related to HCI, without the need for immediate project manager involvement. This empowers UI Designers to quickly iterate based on their expertise in interaction design, streamlining the early stages.
3. **Real-Time Risk Analysis**
    - **Current Issue:** Risk analysis is currently treated as a separate, later stage, which causes delays and does not align with management's expectations.
    - **Change:** Implement **ongoing risk analysis** that evolves in real-time alongside mockup development. The ICT project team should continuously update the risk analysis after every significant design check (e.g., after HCI review and user experience check), not only at the final stages.
4. **Automation and Tooling for Risk and QA Reporting**
    - **Current Issue:** Manual quality control and risk reporting add to the workload and cause delays.
    - **Change:** Introduce automated tools to provide real-time feedback on design issues related to HCI and user experience. Tools like automated usability testing or risk management software can support faster, more accurate assessments. This reduces manual QA efforts and accelerates the feedback loop.
5. **Simplified Decision-Making Process**
    - **Current Issue:** The process requires multiple rounds of approval, creating bottlenecks, especially when adjustments are necessary.
    - **Change:** Delegate decision-making for HCI and UX directly to the UI Designers and Frontend Quality Assurance team. Involve the project manager only when critical changes affect broader project goals or introduce significant risks.

Updated Optimized Process Flow:

1. **Create technical solution concept** → no change.
2. **Develop mockup** → Use placeholder wording in early mockups, allowing designers to focus on HCI without translation.
3. **Mockup Review: Human-Computer Interaction** → UI Designers take full responsibility for this stage, making quick iterative improvements.
4. **Design Check: User Experience** → Continue involving QA and UI Designers; project manager only steps in for major design risks.
5. **Ongoing Risk Analysis** → The ICT project team should perform a continuous risk analysis after each review, updating management proactively.
6. **Define Final Wording and Translations (if necessary)** → Involve the translation team only at this stage to minimize their workload.
7. **Implement feedback and adapt translations if needed** → Quick iterations with final wording.
8. **Freeze mockup** → Conduct final risk analysis updates before freezing.

9. **Design Review: Concept and technical feasibility** → Use automated QA tools for the final review.

Rationale for the Changes:

- **Translation Team’s Involvement:** Involving the translation team only at the end when necessary, allows for a faster design process while reducing their workload.
- **HCI Prioritization:** By giving UI Designers autonomy over HCI-related decisions, you eliminate unnecessary feedback loops, allowing for faster progression.
- **Continuous Risk Analysis:** Integrating real-time risk analysis ensures management’s expectations are met without waiting for the final stages, and decisions can be made with evolving risks in mind.
- **Automation:** Automation for quality assurance and risk feedback ensures the process runs smoothly without extensive manual intervention.

This optimized process is better aligned with the project goals while considering resource constraints.

LLM-Output 23: Context-enhanced Design Process Optimization (Bitmap) R4

R5

To optimize the design process in your flowchart, I would recommend the following changes for efficiency, clarity, and communication flow improvement:

1. **Merge Repetitive Steps**
   - **Current Issue:** The steps “Define Wording and create Translations of mockup elements” and “Adapt Wording and Translations to align them with changed mockup” can be streamlined.
   - **Change:** Combine both translation-related steps into a single "Translate and adapt wording as mockup evolves.” By integrating translation adjustments directly into the mockup review phases (before freezing), you reduce back-and-forth steps.
2. **Parallel Tasks for Speed**
   - **Current Issue:** Instead of a strictly linear progression, certain tasks could occur in parallel.
   - **Change:** Allow translation work to proceed in parallel during mockup development and QA phases. For example, while the ICT Project Team works on technical adjustments based on feedback, the Translation Team can begin refining the glossary and list of translations to align with evolving elements.
3. **Early Feedback Integration**
   - **Current Issue:** Feedback from the Translation Team occurs too late in the process, causing potential rework.
   - **Change:** Add "Early translation feedback” after the mockup review stage to reduce late-stage translation changes. This will identify issues with terminology early in the process.
4. **Combine QA Report Stages**
   - **Current Issue:** The QA reports for both Human-Computer Interaction and User Experience are separate, creating duplication.
   - **Change:** Merge “Mockup Review: Human-Computer Interaction” and “Design Check: User Experience” into a single step—"Mockup and User Experience Review.” This makes the process leaner and minimizes duplication of effort.
5. **Freeze Mockup Earlier**
   - **Current Issue:** Constant mockup adjustments after feedback create delays.
   - **Change:** Freeze the mockup after the QA/feedback phase, before final translations, with strict rules for future changes. This streamlines the process and avoids over-revision.

6. **Risk Analysis in Parallel with Mockup Freezing**
   - **Current Issue:** Risk analysis occurs after the mockup freeze, extending the timeline.
   - **Change:** Allow risk analysis updates to occur concurrently with the mockup freezing stage, rather than afterward, to save time.

Optimized Workflow

1. **Start.**
2. **Create technical solution concept.**
   - *Participating:* Project Manager, ICT Project Team.
   - *Documents:* Requirements Specification, Development Guidelines, Design Document.
3. **Develop mockup.**
   - *Participating:* Project Manager, ICT Project Team, UI Designer.
   - *Documents:* Design Document, Mockup.
4. **Mockup and User Experience Review.**
   - *Participating:* Project Manager, UI Designer, QA Coordinator.
   - *Documents:* Mockup, QA Report.
5. **Early Translation Feedback and Adjustments.**
   - *Participating:* Translation Team.
   - *Documents:* Glossary, List of Translations.
6. **Implement feedback and finalize mockup (Includes translation and feedback incorporation).**
   - *Participating:* ICT Project Team, UI Designer, Translation Team.
   - *Documents:* QA Report, Mockup, Glossary.
7. **Freeze Mockup & Update Risk Analysis (in parallel).**
   - *Participating:* Project Manager, ICT Project Team.
   - *Documents:* Mockup, Risk Analysis, Design Document.
8. **Design Review: Concept and technical feasibility.**
   - *Participating:* Project Manager, Coordinator, Review Team.
   - *Documents:* IT Service Tree, Risk Analysis, Design Document.
9. **End.**

Key Changes Summary

- **Merged steps:** Combined translation and QA-related tasks to avoid redundancies.
- **Parallel tasks:** Performed translation and risk analysis steps in parallel with mockup development to save time.
- **Earlier feedback:** Introduced translation team input earlier in the process to reduce rework later.

This optimization reduces unnecessary iterations and speeds up the design process by promoting parallel work and better communication across teams.

LLM-Output 24: Design Process Optimization (Bitmap) R5

Based on the given requirements, here's a further optimized version of the process flow to address the priorities for Human-Computer Interaction, minimize the translation team's workload, and incorporate ongoing risk analysis.

Optimized Process Workflow:

1. **Start**.
2. **Create technical solution concept**.
    - *Responsible:* Project Manager, ICT Project Team.
    - *Documents:* Requirements Specification, Development Guidelines, Design Document.
3. **Develop initial mockup (focus on HCI)**.
    - *Responsible:* Project Manager, UI Designer.
    - *Participating:* ICT Project Team.
    - *Documents:* Design Document, Mockup.
    - *Notes:* Since wording isn't a priority, focus on functional and interaction elements rather than perfect wording.
4. **Mockup Review: Human-Computer Interaction (HCI)**.
    - *Responsible:* Project Manager.
    - *Participating:* UI Designer, QA Coordinator.
    - *Documents:* Mockup, QA Report.
    - *Notes:* Assess HCI features without involving translation team unless absolutely necessary for interaction-related text. This minimizes their involvement in non-crucial phases.
5. **Design Check: User Experience (optional, if needed)**.
    - *Responsible:* Project Manager.
    - *Participating:* UI Designer, QA Coordinator.
    - *Documents:* Mockup, QA Report.
    - *Notes:* If needed, the wording is reviewed for UX purposes, but only if it impacts usability significantly.
6. **Ongoing Risk Analysis (Parallel Step)**.
    - *Responsible:* Project Manager.
    - *Participating:* Subject Experts, ICT Project Team.
    - *Documents:* Risk Analysis, QA Report.
    - *Notes:* Risk analysis is continuously updated throughout the process as the mockup evolves, rather than waiting until the final review phase.
7. **Integrate Feedback into Mockup**.
    - *Responsible:* ICT Project Team.
    - *Participating:* Project Manager.
    - *Documents:* QA Report, Mockup.
    - *Notes:* Minor wording issues can be left for later unless critical to the design or user interaction.
8. **Wording Adjustments (Only when necessary)**.
    - *Responsible:* UI Designer.
    - *Participating:* Translation Team (if needed).
    - *Documents:* Mockup, List of Translations, Glossary.
    - *Notes:* If wording impacts interaction or UX critically, translation team involvement is minimal and selective to avoid overburdening.
9. **Freeze Mockup**.

  - *Responsible:* ICT Project Team.
  - *Documents:* Mockup, Risk Analysis.
  - *Notes:* Freeze mockup after critical feedback is integrated, focusing on HCI and minimizing non-essential wording updates.

10. **Finalize Risk Analysis and Design Review**.

    - *Responsible:* Project Manager, Review Team.
    - *Documents:* IT Service Tree, Risk Analysis, Design Document.
    - *Notes:* Finalize risk analysis in parallel with the mockup freezing phase to meet management's expectations for real-time updates.

11. **End**.

Key Adjustments Based on the New Criteria:

- **Translation Team Workload Minimized:** Involvement of the translation team is reduced to critical moments where wording impacts interaction or UX significantly. This ensures that their workload remains manageable.

- **Risk Analysis is Continuous:** Risk analysis runs parallel to the design and mockup stages, evolving together with the project to meet management's requirements for real-time updates.

- **Focus on Interaction Over Wording:** Mockups prioritize Human-Computer Interaction (HCI), and wording is only refined where necessary, reducing unnecessary adjustments and translation iterations.

This streamlined approach ensures alignment with management's expectations for evolving risk analysis while maintaining focus on interaction, minimizing wording tasks, and reducing the translation team's workload.

LLM-Output 25: Context-enhanced Design Process Optimization (Bitmap) R5

Implementation Process

R1

To optimize the process described in the flowchart, I would propose the following changes to improve efficiency, reduce redundancy, and enhance clarity. Below are the specific optimizations I would make, followed by the reasoning behind each change:

1. **Merge "Create Test Checklist" and "Manually Perform Functional and Unit Tests"**
   Change: Combine these two steps into one broader task called "Prepare and Execute Test Plan."
   Reason: These two steps seem closely related, as the creation of a test checklist and the execution of tests are typically part of a single iterative process. By merging them, you reduce the transition time between phases and allow the test plan to evolve dynamically during testing.

2. **Concurrent Code Review and Testing**
   Change: Allow the code review (secure coding) to be performed in parallel with testing once the functionality is implemented.
   Reason: Code reviews and functional tests serve different purposes but can be done concurrently. This reduces the overall time required for the project since the review team doesn't need to wait for the full testing phase to be completed.

3. **Automate Testing Where Possible**
   Change: Add a step or annotation in the flowchart that highlights the possibility of automated testing (such as unit tests or continuous integration tests) where appropriate.
   Reason: Automation reduces manual testing effort and increases efficiency by enabling faster identification of issues without waiting for full manual test cycles.

4. **Document Before Code Review**
   Change: Move the "Document functionality" step before the "Code review" step.
   Reason: Documenting functionality before the code review helps the reviewers better understand the system's behavior and objectives, ensuring that the code aligns with documented functionality.

5. **Parallelize Stakeholder Testing**
   Change: Allow stakeholder testing to begin in parallel with the final testing stages, once major functionalities and core requirements have passed internal testing.
   Reason: This shortens the overall project timeline by engaging stakeholders sooner and ensuring that any user-specific requirements are caught early, reducing last-minute changes.

6. **Add Continuous Integration/Continuous Deployment (CI/CD)**
   Change: Introduce a step for CI/CD integration to ensure that functionality and tests are continuously updated and deployed.
   Reason: Implementing CI/CD ensures that the software is always in a deployable state and helps catch integration issues early.

7. **Simplify the Decision Point**
   Change: Combine the decision points regarding successful testing and fulfilled requirements into a single question: "Are all tests and requirements met?"
   Reason: This reduces the number of decision points and eliminates a potential rework loop, making the process more streamlined.

8. **Risk Assessment Before Testing**
   Change: Add a step before "Create Test Checklist" for a brief risk assessment.
   Reason: Identifying risks before testing can help focus on critical areas and avoid unnecessary delays in later stages. This also ensures that the most important risks are addressed first.

Final Optimized Flow:

1. Start

2. Create versioned source code project

3. Implement functionality

4. **Prepare and execute test plan** (merged checklist creation and testing, includes manual and automated tests)

5. **Are all tests and requirements met?** (combined decision point)

   - No → Iterate on test execution and functionality improvement
   - Yes → Proceed

6. Document functionality (moved earlier)
7. Code review (secure coding, concurrent with stakeholder review)
8. Create test plan for stakeholders (parallelized)
9. **Are all tests and requirements met?** (final verification)
   - Yes → End

These changes aim to reduce rework, improve clarity, and streamline the process by allowing for more parallel tasks.

LLM-Output 26: Implementation Process Optimization (Bitmap) R1

Considering your organization's **"Just-in-Time Documentation"** approach and the fact that you do not follow **"Test-Driven Development" (TDD)**, I will adjust the previously suggested optimizations to align with these principles:

1. **Document After Code Review (Just-in-Time Documentation)**
   - **Change**: Shift the "Document Functionality" step to occur **after** code review and successful internal testing.
   - **Reason**: Since your organization follows the Just-in-Time Documentation approach, it's best to ensure the documentation reflects the **final, stable implementation**. This avoids rework caused by changes made during the testing and code review phases.
2. **No Pre-Test Documentation or Test-Driven Development**
   - **Change**: Maintain the sequence of development followed by testing, with no premature test case creation before implementation.
   - **Reason**: Since TDD is not part of your methodology, we should avoid introducing test cases too early. The tests should be based on the **finalized functionality** after it's implemented, following your organization's approach to avoid bugs due to incomplete requirements.

Updated Optimization Proposal:

1. **Merge Test Checklist Creation and Manual Testing**
   - **Change**: Combine "Create Test Checklist" and "Manually Perform Functional and Unit Tests" into a broader step called "Prepare and Execute Test Plan."
   - **Reason**: Test creation and execution should evolve based on the functionality developed, in line with your non-TDD approach. The testing process should reflect the actual implementation without the need for upfront test cases.
2. **Concurrent Code Review and Testing**
   - **Change**: Allow code reviews (secure coding) to happen **in parallel** with testing.
   - **Reason**: This saves time and ensures the code can be reviewed while some tests are still ongoing, aligning with the iterative nature of your process.
3. **Prioritize Automated Testing to Reduce Manual Load**
   - **Change**: Emphasize automation wherever possible during testing.
   - **Reason**: While the focus remains on manual testing, automating repetitive tasks (like unit tests) can help ensure efficiency and reduce unnecessary manual effort without the constraints of TDD.
4. **Post-Test Documentation**

   - **Change**: Move documentation to occur **after all tests and code reviews** are completed.
   - **Reason**: This aligns with your Just-in-Time Documentation approach, ensuring that documentation is accurate and up to date with the final version of the product.

5. **Real-Time Feedback Integration**
   - **Change**: Introduce real-time feedback integration from stakeholders or key team members during the later testing stages.
   - **Reason**: Getting feedback during testing can help refine the product without needing to stop for formal stakeholder testing phases, minimizing delays and rework.

6. **Focus on Risk-Based Testing**
   - **Change**: Prioritize testing critical functionalities based on risk assessment, especially for areas where incomplete requirements could introduce bugs.
   - **Reason**: Without TDD, it's essential to focus testing efforts on high-risk areas, ensuring critical functionalities are thoroughly tested without prematurely creating tests for less critical parts.

Optimized Process Flow:

1. **Start**

2. **Create versioned source code project**
   - Responsible: Project Manager, ICT Project Team

3. **Implement functionality**
   - Responsible: ICT Project Team
   - Participating: ICT Project Team

4. **Prepare and execute test plan**
   - Responsible: ICT Project Team
   - Participating: ICT Project Team
   - **Includes test checklist creation and execution (manual/automated)**

5. **Tests successful?**
   - **No** → Iterate on tests and functionality
   - **Yes** → Proceed

6. **Code review (secure coding)** (parallel with tests)
   - Responsible: Project Manager, Coordinator Review Team

7. **Document functionality** (post-tests and review)
   - Responsible: ICT Project Team
   - Participating: Project Manager
   - **Just-in-Time Documentation** based on final implementation

8. **Create test plan for stakeholders**
   - Responsible: ICT Project Team
   - Participating: Project Manager, Stakeholders

9. **All requirements fulfilled?**
   - Responsible: Project Manager, Stakeholders
   - Participating: ICT Project Team
   - **No** → Iterate on feedback and testing

- **Yes** → End

Key Considerations:

- **Documentation occurs only after** all tests and reviews are finalized to align with the "Just-in-Time Documentation" approach.
- **Test checklist creation and execution** are merged into one step, reducing redundancy and avoiding the TDD approach.
- Testing focuses on **real-time risk assessment** without pre-creating test cases, reducing the risk of incomplete or incorrect requirements.

This flow ensures that testing reflects actual functionality, and documentation is accurate and up to date with minimal rework, while also keeping efficiency in mind. Let me know if you'd like me to refine any specific area further or adjust this process more!

LLM-Output 27: Context-enhanced Implementation Process Optimization (Bitmap) R1

R2

To optimize this process flow, several improvements can be considered. Here's an outline of the changes, with a rationale for each:

Key Changes:

1. **Parallelizing Tasks**
   Currently, the process is mostly sequential. Some tasks, like creating documentation and preparing the test checklist, could be done in parallel to save time.
   Example: While the team is performing the functional and unit tests, documentation work could be started.
2. **Automating Functional and Unit Tests**
   Change: Introduce automation for functional and unit tests. Automated tests can reduce human error, speed up the process, and allow for quicker feedback.
   Reason: Manual testing is time-consuming and prone to mistakes.
3. **Feedback Loops**
   Change: Add a feedback loop after the "Create Test Plan for Stakeholders" step.
   Reason: Involving stakeholders earlier in the process could provide valuable feedback, avoiding potential rework later. It helps validate assumptions before completing the entire cycle.
4. **Code Review Enhancement**
   Change: Implement automated code review tools alongside manual reviews.
   Reason: This can speed up the process while ensuring secure coding practices through automated checks, reducing the burden on manual reviewers.
5. **Risk Management and Continuous Integration**
   Change: Introduce continuous integration (CI) and continuous risk assessment at multiple points in the process, especially before the code review step.
   Reason: Continuous integration tools can help catch errors earlier and ensure that new code doesn't break existing functionality.
6. **Requirement Validation Throughout**
   Change: Instead of checking if "all requirements fulfilled" at the very end, implement periodic requirement validation checkpoints after key steps like "Implement Functionality" and "Test Checklist Creation."
   Reason: This ensures that the development aligns with requirements continuously, reducing last-minute surprises.

Optimized Process Overview:

1. **Start**
2. **Create Versioned Source Code Project**

- No change here.

3. **Implement Functionality**
    - Implement Continuous Integration (CI).
4. **Create Test Checklist**
    - Start creating documentation in parallel.
5. **Manually Perform Functional and Unit Tests**
    - Use automated testing tools alongside manual tests.
    - Parallel task: Start preparing stakeholder documentation.
6. **Tests Successful?**
    - **Yes**: Proceed to code review.
    - **No**: Go back to implementation with automated feedback on failed tests.
7. **Code Review (Secure Coding)**
    - Incorporate automated code review tools.
8. **Document Functionality**
    - Parallel task: Update the test checklist and align with new requirements.
9. **Create Test Plan for Stakeholders**
    - Introduce a stakeholder feedback loop here.
10. **All Requirements Fulfilled?**
    - If requirements are not fulfilled, feedback should be addressed in the previous stages.
11. **End**

These changes aim to make the process faster, more efficient, and reduce the risk of rework by allowing earlier feedback from stakeholders and automating repetitive tasks.

LLM-Output 28: Implementation Process Optimization (Bitmap) R2

Given the information about your organization's emphasis on **"Just-in-Time Documentation"** and the decision not to follow **"Test-Driven Development"**, here's an updated optimization that aligns with your approach:
Revised Key Changes:

1. **Just-in-Time Documentation (JITD)**:
    - **Change**: Documentation tasks will only begin after the functionality is fully implemented and stabilized. Documentation work would happen after the successful test phase to reflect the final implementation.
    - **Reason**: This avoids the risk of reworking documentation based on unfinished or changing functionality, aligning with your JITD approach.
2. **Post-Development Testing**:
    - **Change**: Since you're not using Test-Driven Development (TDD), the creation of test checklists and the execution of tests will happen only after the implementation is complete, without early test case creation.

- **Reason**: This ensures that test cases are aligned with finalized code, reducing the chance of missing key requirements due to premature test definition.

3. **Maintaining Manual Functional and Unit Tests**:
   - **Change**: Continue to perform functional and unit tests manually, but integrate tools for semi-automation to expedite repetitive tasks (e.g., data validation, user interface checks).
   - **Reason**: Since TDD is not practiced, having rigorous manual testing ensures key requirements are not missed, but semi-automation helps boost efficiency without compromising accuracy.

4. **Deferring Stakeholder Feedback**:
   - **Change**: Instead of involving stakeholders too early, keep the process focused on implementation and internal testing before preparing the final test plan and review documentation.
   - **Reason**: This reduces unnecessary back-and-forth with stakeholders during the iterative development stages, reflecting your efficiency-driven approach.

Updated Optimized Process:

1. **Start**

2. **Create Versioned Source Code Project**
   - No change.

3. **Implement Functionality**
   - Focus on getting the core implementation done first before involving documentation or stakeholder feedback.

4. **Create Test Checklist**
   - Create the test checklist after functionality is implemented.

5. **Manually Perform Functional and Unit Tests**
   - Perform rigorous manual testing, with possible semi-automation for repetitive tasks.

6. **Tests Successful?**
   - **Yes**: Proceed to code review.
   - **No**: Go back to refine the implementation.

7. **Code Review (Secure Coding)**
   - Continue with manual reviews with secure coding in mind.

8. **Document Functionality (JITD)**
   - Start documenting only after successful code review and tests, ensuring that documentation reflects the final, stable implementation.

9. **Create Test Plan for Stakeholders**
   - Finalize the test plan based on the reviewed and tested code.

10. **All Requirements Fulfilled?**
    - If yes, proceed to finalize the project.

11. **End**

This version of the process keeps your workflow efficient, minimizes unnecessary rework, and ensures that the final documentation and tests align with the completed, stable codebase.

LLM-Output 29: Context-enhanced Implementation Process Optimization (Bitmap) R2

R3

To optimize the process depicted in your flowchart, here are some suggestions along with changes that could streamline the process while maintaining quality and thoroughness.

Optimizations and Changes:

1. **Consolidation of Testing Phases**
   Change: Combine the “Create Test Checklist” and “Manually Perform Functional and Unit Tests” phases.
   Rationale: Since test checklists and test execution are tightly coupled, having them in one step could reduce delays between checklist creation and test execution, as both often involve the same resources. This ensures testing starts immediately after checklist preparation.

2. **Automated Testing Integration**
   Change: Introduce an "Automated Testing" step after “Manually Perform Functional and Unit Tests."
   Rationale: Implementing automated testing (especially for unit tests) could significantly reduce the manual testing burden and increase test coverage. This improves efficiency while ensuring frequent checks on code quality.

3. **Parallelize Documentation and Code Review**
   Change: Allow "Document Functionality" and "Code Review" to happen in parallel.
   Rationale: Documenting the functionality can be done independently by the person(s) responsible for the feature while the code review occurs, thus reducing downtime and speeding up the process.

4. **Add Continuous Integration (CI)**
   Change: Add a “CI Integration” step before "Create Test Plan for Stakeholders."
   Rationale: Continuous integration ensures that all code changes are regularly tested, which helps in identifying issues early and maintaining a stable development environment before the stakeholder test phase.

5. **Feedback Loop in Stakeholder Testing**
   Change: Add a feedback loop after “Create Test Plan for Stakeholders” to "Implement Functionality."
   Rationale: If stakeholder testing reveals that requirements are not fulfilled, this direct loop allows the ICT team to immediately address any functional gaps, ensuring that stakeholder needs are met without restarting the entire process.

6. **Risk Analysis Continuously**
   Change: Introduce a continuous “Risk Analysis” phase throughout the process, from the start to the end.
   Rationale: Integrating risk analysis early and consistently at every stage helps identify and mitigate potential issues sooner, avoiding costly rework.

Revised Workflow:

1. **Start**

2. **Create Versioned Source Code Project**

3. **Implement Functionality**

4. **Create & Perform Tests (Functional, Unit, and Automated)**

5. **Test Successful?**

   - **No**: Go back to "Implement Functionality."
   - **Yes**: Continue.

6. **Document Functionality** (Parallelized with step 7)

7. **Code Review (Secure Coding)** (Parallelized with step 6)

8. **CI Integration**

9. **Create Test Plan for Stakeholders**

10. **Stakeholder Tests: All Requirements Fulfilled?**

    - **No**: Feedback loop to "Implement Functionality."
    - **Yes**: End.

This approach minimizes bottlenecks, maximizes parallelization where applicable, and integrates modern development practices like CI and automated testing for improved efficiency and quality control.

LLM-Output 30: Implementation Process Optimization (Bitmap) R3

Considering your organization's use of **"Just-in-Time Documentation"** and avoidance of **"Test-Driven Development" (TDD)**, here's an optimized process tailored to these principles:
Key Adjustments:

1. **Just-in-Time Documentation:**
   - **Change**: Move the "Document Functionality" step to the end of the process after all functionality has been finalized and tested.
   - **Rationale**: This avoids any rework of documentation by ensuring that it is created only after the implementation is fully complete. This aligns perfectly with your organization's approach to documentation, reflecting the final state of the system without unnecessary intermediate drafts.
2. **Shift Focus from Test-Driven Development (TDD):**
   - **Change**: The process should focus on implementing functionality first without creating test cases prematurely. Ensure tests are created only after a clear understanding of the requirements and implementation.
   - **Rationale**: This reduces the risk of premature tests missing important requirements or becoming outdated as the project evolves.

Updated Process:

1. **Start**
2. **Create Versioned Source Code Project**
3. **Implement Functionality** (Based on clear and finalized requirements)
4. **Create Test Checklist & Perform Functional/Unit Tests** (Automated tests where applicable)
5. **Test Successful?**
   - **No**: Return to "Implement Functionality"
   - **Yes**: Continue
6. **Code Review (Secure Coding)**
7. **CI Integration (if applicable)**
8. **Create Test Plan for Stakeholders**
9. **Stakeholder Testing: All requirements fulfilled?**
   - **No**: Feedback loop to "Implement Functionality"
   - **Yes**: Proceed to next step
10. **Document Final Functionality** (Just-in-Time Documentation reflecting the final state)
11. **End**

Rationale for Key Adjustments:

- **Just-in-Time Documentation**: Moving the documentation phase to the end ensures that the documentation effort is efficient, avoids rework, and perfectly matches the final implementation.
- **Focus on Post-Implementation Testing**: Since TDD is not followed, the testing steps (checklist and actual tests) are created after the implementation, ensuring that the testing is relevant and robust without the risks of missing critical functionality.

This approach ensures that documentation aligns with your organization's priorities, while the absence of TDD is balanced by strong post-implementation testing.

LLM-Output 31: Context-enhanced Implementation Process Optimization (Bitmap) R3

R4

To optimize this process, I will suggest some improvements based on common best practices and areas where efficiency can be enhanced:

1. **Parallelization of Tasks**
   Code Review and Test Checklist: Code review can start after implementing functionality, rather than waiting until after tests have passed. This will reduce idle time for the review team, as they can review parts of the functionality already implemented before testing is fully completed.
   Test Plan Creation: Rather than waiting until after documentation is complete to start the test plan creation, this task can run in parallel once the implementation phase is finished, based on the test checklist.

2. **Automated Testing Integration**
   Manual Unit and Functional Tests: Introduce automated testing to reduce the time and effort spent on manual testing. Automated test suites can quickly run through large numbers of tests and ensure that code is regression-free.

3. **Continuous Testing**
   Test after Each Functional Addition: Instead of waiting for the full implementation before testing, introduce continuous testing. This means running tests after every significant functionality addition. This can lead to quicker identification and resolution of bugs.

4. **Documentation Updates During Development**
   Concurrent Documentation: Allow for partial documentation to be updated during the implementation phase rather than waiting until functionality is fully implemented. This allows the documentation team to keep pace with the development, which will speed up the overall process.

5. **Risk Analysis Integration**
   Risk analysis should be part of each step of the process, especially during the testing and code review stages. This ensures that security, performance, and scalability risks are addressed early, avoiding costly revisions later.

6. **Stakeholder Involvement at Early Stages**
   Early Feedback: Involve stakeholders earlier in the process, potentially right after a significant functional module is implemented, to ensure alignment and reduce any surprises in later stages. This feedback loop can be invaluable and prevent large-scale reworks.

Revised Process Flow:

- **Implementation & Continuous Testing**:
  - Implement functionality → Partial code review → Continuous testing of new functionality (automated or manual) → Parallel update of the test plan and documentation.
- **Final Stages**:
  - Once the final feature set is implemented and tested, a full code review, followed by final testing and documentation, can be completed.
- **Review & Feedback**:
  - Include stakeholders in feedback sessions after each major module's implementation, thus aligning their requirements continuously.

Benefits of these Changes:

- **Time Efficiency**: Reducing waiting times by running tasks in parallel and involving automated tools for testing.
- **Early Detection**: Problems and risks are identified and mitigated earlier in the process.
- **Reduced Manual Workload**: Automating tests cuts down the time spent on manual verification.
- **Stakeholder Satisfaction**: Continuous alignment ensures that stakeholder needs are met earlier, reducing rework at the end.

Would you like me to draw a revised process flow diagram with these suggestions?

LLM-Output 32: Implementation Process Optimization (Bitmap) R4

Taking into account your organization's approach of **Just-in-Time Documentation** and not following **Test-Driven Development (TDD)**, here are the tailored optimizations:

1. **Just-in-Time Documentation Compliance**:
   - **Documentation at Final Implementation**: Since documentation reflects the final product, it should only begin after the core functionality is fully implemented and confirmed to work. In this case, rather than concurrent documentation updates during development, the documentation will occur after the successful tests, once the implementation is stable and final. This ensures that the documentation will match the final product without needing multiple revisions.
2. **Risk-Adjusted Testing without TDD**:
   - **Functional and Unit Testing (Manual & Automated)**: Since your organization avoids TDD, it's important to still catch bugs early without overly relying on pre-built tests. While automated testing remains valuable, I suggest focusing heavily on **manual exploratory testing** after each functional block is completed. This ensures that key requirements are fully understood before test cases are created.
3. **Parallelization in Non-Testing Phases**:
   - **Code Review After Initial Testing**: Instead of starting code reviews during the implementation phase (which is more in line with TDD), it should occur after passing initial functional and unit tests. This ensures that the code review team works only with near-final code, reducing rework during the review process.
4. **Flexibility in Stakeholder Feedback**:
   - Since early stakeholder feedback may lead to changes, it's important to review requirements carefully before testing. You could still involve stakeholders in **early checkpoints** after partial functionality is stable, but the feedback would primarily focus on alignment rather than final approval, maintaining flexibility.

Optimized Process Flow (Adjustments):

- **Post-Implementation Documentation**:
  - Implement functionality → Functional/unit tests → Code review (secure coding) → Final documentation.
- **Manual and Exploratory Testing Focus**:
  - Implement functionality → Create test checklist → Perform manual tests (functional & unit) to assess if key requirements are met → Successful tests? If **No**, refine functionality, otherwise proceed.
- **Test Plan Creation**:
  - The test plan for stakeholders and external users can begin after successful internal tests but before the final review, ensuring alignment with the final documentation.

Benefits:

- **Avoid Rework in Documentation**: Documentation only happens after the functionality is stable and final, eliminating any need for multiple iterations.
- **Higher Quality Code Reviews**: Code reviews happen on fully tested, stable code, reducing back-and-forth changes during review.
- **Manual Testing Emphasis**: This approach ensures that edge cases and critical requirements are not overlooked by automated scripts.
- **Efficient Stakeholder Involvement**: Feedback loops are timely, ensuring the project stays aligned with requirements without unnecessary delays.

This approach fits your documentation strategy and avoids premature test creation without compromising on efficiency or quality.
Would you like further refinements or visual representation of this flow?

LLM-Output 33: Context-enhanced Implementation Process Optimization (Bitmap) R4

R5

To optimize this process, we can focus on improving efficiency, reducing redundancies, and enhancing communication between stakeholders. Here's an optimized version and the rationale behind each change:

1. **Parallel Work for Functional Implementation and Testing**
   Change: Allow parallel work between implementing functionality and creating the test checklist. This eliminates waiting time and improves efficiency.
   Reason: While functionality is being developed, the test checklist can be prepared based on the specification sheet. This way, once the functionality is complete, tests can start immediately, saving time.

2. **Automated Testing Integration**
   Change: Add automated testing between "Manually Perform functional and unit tests" and "Tests successful?" for routine checks.
   Reason: Automating certain repetitive and predictable tests (like unit tests) speeds up the testing process and increases accuracy by reducing human errors. Manual testing can still be performed but should focus on edge cases and functionality validation.

3. **Conditional Gate for Code Review**
   Change: Add a condition before "Code Review (secure coding)" that checks if the implementation impacts security-critical features. Only proceed with a secure code review if needed.
   Reason: Not all code changes impact security. This step can sometimes be skipped or deferred, depending on the scope of the change, reducing unnecessary delays.

4. **Simultaneous Documentation**
   Change: Allow documentation of functionality to begin as soon as functionality is implemented, instead of waiting until after testing is successful.
   Reason: Documentation is often delayed until testing is done, but non-technical documentation can begin earlier, reducing the overall timeline. Developers can provide technical input as the code progresses, refining documentation iteratively.

5. **Continuous Feedback Loop with Stakeholders**
   Change: Add a stakeholder feedback loop after creating the test plan for stakeholders.
   Reason: Early involvement of stakeholders allows for faster feedback and ensures that requirements are met without unnecessary back-and-forth at the end of the process. This can reduce the need for extensive rework later on.

6. **Clarifying Requirements Verification**
   Change: Explicitly specify that verification of "all requirements fulfilled" should be both functional and non-functional (e.g., performance, security, usability).
   Reason: Sometimes, only functional requirements are considered. Non-functional requirements are equally important, and their inclusion ensures a more robust final product.

7. **Risk-based Testing Approach**
   Change: Implement a risk-based testing approach by adding a "Risk Analysis" phase before performing tests.
   Reason: Prioritizing testing based on risk can ensure that the most critical parts of the system are tested thoroughly, while less critical components receive less intensive testing, saving time and resources.

Updated Process Flow:

1. **Start**

2. **Creating a versioned source code project** → (Version management)

3. **Parallel**:

   - **Implement Functionality** → (Specification Sheet, Version management)
   - **Create Test Checklist** → (Specification Sheet, Version management, Internal Test Checklist)

4. **Automated Tests (if applicable)** → Proceed based on result

5. **Manually Perform functional and unit tests** → (Specification Sheet, Version management, Internal Test Checklist)

6. **Tests successful?**

- **No** → Implement fixes
- **Yes** → **Proceed**

7. **Conditional Secure Code Review (if security-relevant)**:
    - If necessary → (Version management, Code Review Report)
8. **Document Functionality** (can begin earlier, parallel to some testing) → (User Documentation)
9. **Create Test Plan for Stakeholders** → (User Documentation, External Test Checklist)
10. **Stakeholder feedback loop** → Ensures alignment with expectations
11. **All requirements fulfilled?** (Functional and Non-functional)
    - **Yes** → **End**

Would you like to see this updated process flow in diagram form?

LLM-Output 34: Implementation Process Optimization (Bitmap) R5

Given your organization's focus on efficiency through **"Just-in-Time Documentation"** and the avoidance of **"Test-Driven Development" (TDD)**, we can further tailor the optimizations to align with these principles:
Adjusted Optimizations:

1. **Documentation Alignment with Just-in-Time Approach**:
    - **Change**: Remove the suggestion to start documentation in parallel with functionality implementation.
    - **Reason**: To respect your "Just-in-Time Documentation" approach, documentation should only be created after the functionality and testing are completed to avoid rework.
2. **Refined Testing Without TDD**:
    - **Change**: Maintain the existing flow of creating the test checklist *after* functionality is implemented.
    - **Reason**: Since TDD is not followed, we maintain a more reactive testing strategy where test cases are based on completed features. This minimizes the risk of missing key requirements during the development phase.
3. **Iterative Test Development (Post-Implementation)**:
    - **Change**: After the functionality is implemented, the test checklist can be adjusted iteratively based on feedback from manual and automated testing.
    - **Reason**: This approach ensures that test cases are designed based on the final product rather than hypothetical features, further preventing unnecessary work.

Updated Process Flow Adjustments:

1. **Start**
2. **Creating a versioned source code project** → (Version management)
3. **Implement Functionality** → (Specification Sheet, Version management)
4. **Create Test Checklist** → (Specification Sheet, Version management, Internal Test Checklist)
5. **Automated Tests (if applicable)** → Proceed based on result
6. **Manually Perform functional and unit tests** → (Specification Sheet, Version management, Internal Test Checklist)

7. **Tests successful?**
   - **No** → Implement fixes
   - **Yes** → Proceed
8. **Conditional Secure Code Review (if security-relevant)**
   - If necessary → (Version management, Code Review Report)
9. **Document Functionality (Post-testing)** → (User Documentation)
10. **Create Test Plan for Stakeholders** → (User Documentation, External Test Checklist)
11. **Stakeholder feedback loop** → Ensures alignment with expectations
12. **All requirements fulfilled? (Functional and Non-functional)**
    - **Yes** → **End**

This adjusted flow respects the principles of "Just-in-Time Documentation" by focusing on documentation at the end of the process and avoids premature testing strategies, aligning with your organizational preferences.
Let me know if you'd like further refinements or a diagram based on this version!

LLM-Output 35: Context-enhanced Implementation Process Optimization (Bitmap) R5

Go-Live Process

R1

To optimize the process depicted in your diagram, I would suggest a few changes to streamline communication, reduce potential delays, and ensure better stakeholder engagement. Below is an outline of these optimizations and the reasoning behind them:

1. **Preliminary Testing Before Production Deployment**
   - **Change**: Add a preliminary testing phase (like User Acceptance Testing - UAT) before the "Deployment on Production Systems" step.
   - **Reason**: This helps identify issues earlier and ensures a smoother production deployment. The UAT can be performed by key stakeholders and users to verify the system's functionality before full deployment.
2. **Parallel Internal Approval and Final Test**
   - **Change**: Allow "Internal Approval of successful deployment" and the "Final Test" to run in parallel.
   - **Reason**: These two steps can occur simultaneously as the approval might be contingent on the final test results. This reduces bottlenecks, saving time.
3. **Automated User Satisfaction Analysis**
   - **Change**: Replace "Manual User Satisfaction Analysis" with an automated feedback collection process, using online surveys or feedback tools from the live system.
   - **Reason**: Automation will expedite data collection and provide more structured insights into user satisfaction, saving time and resources from manual analysis.
4. **Merge PR Campaign and Project Presentation**
   - **Change**: Combine "Project Presentation to Management" and the "PR Campaign" into a single step where management reviews include a briefing on PR initiatives.
   - **Reason**: If the presentation to management is necessary, the PR strategy can be incorporated to avoid separate presentations, streamlining the flow and reducing redundancy.
5. **Documentation Flow Improvements**
   - **Change**: Consolidate the number of documents referenced, especially between deployment and final testing, into a single unified "Deployment and Testing Report."
   - **Reason**: Instead of maintaining several documents like "IT Service Map" and "Internal Test Checklist," combining these into a single report will simplify documentation handling and make the process more efficient for both reference and approval stages.
6. **Final Review before Go Live**
   - **Change**: Introduce a lightweight "Final Readiness Review" just before the "Go Live" step.
   - **Reason**: A quick final review ensures that any last-minute changes or issues can be addressed before going live, reducing the risk of post-launch problems.
7. **Faster Go Live Announcement**
   - **Change**: Simplify the "Announcement" process during the Go Live phase with automated or pre-planned communications (such as using automated emails or notifications).
   - **Reason**: Having a pre-prepared template or automated announcement system ensures a faster Go Live process without delays from manual communication steps.
8. **Early Involvement of PR Team**
   - **Change**: Engage the PR Team earlier in the process, possibly during the "Internal Approval" stage.
   - **Reason**: This allows for PR strategies to be developed in parallel with the internal approval and testing phases, ensuring the team is fully prepared for communication efforts when deployment is complete.

Summary of Optimized Process Flow

1. Preliminary Test (UAT)
2. Deployment on Production Systems
3. Internal Approval and Final Test (Parallel)
4. PR and Management Presentation (Unified)
5. Automated User Satisfaction Collection
6. Final Readiness Review
7. Go Live (with Automated Announcement)
8. Project Review & Effectiveness Evaluation

These changes aim to make the process more efficient by reducing manual steps, creating parallel paths for approval and testing, automating feedback collection, and merging redundant presentation and campaign efforts. Would you like me to illustrate the optimized flow for better clarity?

LLM-Output 36: Go-Live Process Optimization (Bitmap) R1

Considering your organizational focus on reducing internal controls, parallelizing tasks, and gathering feedback post-Go-Live, here's an optimized process based on your guidelines:

1. **Combine Internal Approval and Final Test**
   - **Change**: Merge "Internal Approval of successful deployment" and "Final Test" into one streamlined task.
   - **Reason**: Since internal approval relies on successful test results, it makes sense to combine these steps into one. This will minimize unnecessary control steps and shorten the process timeline.
2. **Eliminate Manual User Satisfaction Analysis**
   - **Change**: Remove the "Manual User Satisfaction Analysis" step entirely.
   - **Reason**: Since feedback is only gathered post-Go-Live through tickets, this step becomes redundant. Real-time ticket creation from the live system can act as the primary means of gathering user feedback, reducing the need for a separate analysis.
3. **Merge PR Campaign with Go Live**
   - **Change**: Combine the "PR Campaign" and "Go Live" steps into one.
   - **Reason**: By preparing the PR communication in parallel with the final stages of deployment and testing, the Go Live and campaign launch can occur simultaneously. This prevents delays and makes the process more efficient by removing the need for separate campaigns after Go Live.
4. **Parallelized Project Presentation and PR Campaign**
   - **Change**: If the "Project Presentation to Management" is needed, run it in parallel with the PR Campaign preparation.
   - **Reason**: Running the management presentation and PR tasks in parallel ensures that both management and public stakeholders are updated without delaying the Go Live phase.
5. **Feedback After Go Live**
   - **Change**: Focus on gathering user feedback solely through post-Go-Live tickets from the live system.
   - **Reason**: Since user satisfaction will be monitored through live system tickets, remove any pre-Go-Live manual feedback collection steps. This will align with your organization's emphasis on collecting feedback only after the system goes live.

Optimized Flow Summary:

- Deployment on Production Systems
- **Combined Final Test & Internal Approval** (Parallel task)
- Project Presentation to Management (if needed) (Parallel to PR)
- **PR Campaign and Go Live** (Combined task)
- Feedback via tickets post-Go Live (replacing Manual User Satisfaction Analysis)
- Project Review & Effectiveness Evaluation

This process focuses on simplifying tasks by combining and parallelizing steps where possible. It also eliminates unnecessary controls and focuses on post-Go-Live feedback, as aligned with your organizational values.

LLM-Output 37: Context-enhanced Go-Live Process Optimization (Bitmap) R1

R2

To optimize the process shown in your image, I would make the following changes:

1. **Simplification of Approval Steps**:
   - **Change**: Merge the "Internal Approval of successful deployment" and "Final Test" into one step called "Final Test and Approval."
   - **Reason**: If the final test is successful, approval can be granted within the same workflow step, eliminating redundancy.
2. **Automate User Satisfaction Analysis**:
   - **Change**: Replace "Manual User Satisfaction Analysis" with "Automated User Satisfaction Collection & Analysis."
   - **Reason**: Automating this step can reduce time and effort while ensuring feedback is gathered consistently from users in real-time.
3. **Decision Point Efficiency**:
   - **Change**: Revise the decision "Project Presentation to Management needed?" to only happen in cases of major changes or risks (e.g., high-risk projects).
   - **Reason**: Streamlining this decision reduces unnecessary meetings for low-impact projects.
4. **Merge PR Strategy and Campaign**:
   - **Change**: Combine the "PR Strategy" and "PR Campaign" into a single task under the PR Team's responsibility.
   - **Reason**: PR strategy planning can happen simultaneously as the campaign is executed, saving time.
5. **Risk Management and Evaluation Improvement**:
   - **Change**: Incorporate "Risk Analysis" earlier, possibly right after deployment to production systems, instead of waiting until the project review.
   - **Reason**: Addressing risks earlier ensures the project can mitigate issues before they impact the end-users.
6. **Increased Documentation Streamlining**:
   - **Change**: Minimize the use of multiple documents by consolidating the "Deployment Plan," "IT Service Map," and "Records of Processing Activities" into one comprehensive "Go-Live Documentation."

- **Reason**: Reduces document management overhead, making it easier for the project team to reference and maintain critical data.

Optimized Workflow Overview:

1. Start
   a) Deployment on Production Systems (with comprehensive Go-Live Documentation)
   b) **Final Test and Approval**
2. Decision: Is a Project Presentation to Management needed?
   - Yes → **Project Presentation to Management**
   - No → **PR Campaign** (PR Strategy embedded)
3. **Automated User Satisfaction Collection & Analysis**
4. **Go Live**
5. **Project Review & Effectiveness Evaluation** (including "Risk Analysis" as a continuous task)

By reducing redundancies, improving automation, and streamlining decision points, this process can become more efficient and scalable.

LLM-Output 38: Go-Live Process Optimization (Bitmap) R1

Given your organization's prioritization of efficiency, minimizing internal controls, and post-Go-Live feedback collection, I would refine the previous optimizations with a stronger emphasis on parallelizing tasks and removing steps that do not directly add stakeholder value.

Updated Process Optimizations:

1. **Parallelize "Final Test" and "Internal Approval"**
   - **Change:** Run the "Final Test" and "Internal Approval of successful deployment" steps in parallel, since both aim to confirm the deployment's success.
   - **Reason:** Parallel execution speeds up the process by reducing delays between validation and approval. This also fits your focus on reducing internal controls that don't add stakeholder value.
2. **Combine User Satisfaction Analysis with Go-Live**
   - **Change:** Remove "Manual User Satisfaction Analysis" as a separate step. Instead, gather user feedback during the "Go Live" phase through automated systems, and log it post-Go-Live for analysis.
   - **Reason:** As tickets are created post-Go-Live, feedback will naturally be collected during the live system phase. There's no need for a separate manual analysis step.
3. **Eliminate Project Presentation Unless Required**
   - **Change:** Remove the "Project Presentation to Management" step unless it is absolutely necessary based on the strategic importance of the project.
   - **Reason:** Only projects that significantly impact stakeholders or management decisions should require a formal presentation, saving time and resources for both the project manager and management.
4. **Run PR Campaign in Parallel with Go-Live Readiness**
   - **Change:** Launch the "PR Campaign" preparation (if needed) in parallel with the "Go Live Readiness Review."
   - **Reason:** This reduces delays between PR preparation and system launch, aligning communication with technical execution.

5. **Integrated Go-Live and Ticket Management**
    - **Change:** Ensure that the "Go Live" step includes an automated system for logging user feedback and tickets, making feedback gathering and resolution part of the Go-Live phase itself.
    - **Reason:** This removes the need for a separate feedback phase and speeds up the response to any issues that arise after the system is live.

Final Optimized Process Flow:

- **Deployment on Production Systems**
- **Parallel: Final Test & Internal Approval**
- **Optional: Project Presentation to Management (only if strategically important)**
- **Parallel: PR Campaign & Go Live Readiness Review**
- **Go Live (includes feedback collection and ticket management)**
- **Project Review & Effectiveness Evaluation**

These changes reduce the overall process timeline by merging and parallelizing tasks where feasible, while maintaining the integrity of the Go-Live feedback and ticketing system, aligned with your organization's efficiency goals.

LLM-Output 39: Context-enhanced Go-Live Process Optimization (Bitmap) R2

R3

To optimize the process outlined in the flowchart, I would recommend the following changes to increase efficiency and streamline tasks:

1. **Internal Approval and Final Test Merging**
    - **Change**: Combine the "Internal Approval of successful deployment" and "Final Test" steps into one, as both involve validating the success of deployment.
    - **Reason**: These steps are closely related and can be performed as part of the same verification process, reducing redundancy.
2. **Automating User Satisfaction Analysis**
    - **Change**: Introduce automated feedback tools or surveys for the "Manual User Satisfaction Analysis" step, instead of relying solely on manual analysis.
    - **Reason**: Automation will save time and allow for more accurate data collection, reducing manual workload and increasing data-driven insights.
3. **PR Campaign Trigger**
    - **Change**: Make the "PR Campaign" step conditional, depending on the nature of the project (internal vs. external). Not all projects may require a PR campaign.
    - **Reason**: Tailoring the PR strategy based on project scope avoids unnecessary work for purely internal projects.
4. **Project Presentation to Management Streamlining**
    - **Change**: Refine the "Project Presentation to Management" process to be an on-demand or required only for specific projects based on the decision criteria (such as project size, strategic impact).
    - **Reason**: This reduces the number of unnecessary presentations for smaller or routine projects, saving time for the project manager and management.
5. **More Focused Go Live Preparation**

- **Change**: Introduce a "Go Live Readiness Review" right before the "Go Live" step, where the project team ensures that all systems and documentation are ready.
- **Reason**: This minimizes the risk of last-minute issues during the actual go-live process, ensuring that all aspects have been checked and are ready for launch.

6. **Centralize Documentation Storage**
    - **Change**: Integrate all documents into a central, easily accessible repository (e.g., all "Documents/-Data" such as Deployment Plans, PR Strategies, etc. in one location).
    - **Reason**: Having a centralized location for all documents reduces time spent searching for and managing different files.

Final Optimized Flow:

1. Deployment on Production Systems
2. Combined: Final Test & Internal Approval
3. Conditional: Project Presentation to Management
4. Conditional: PR Campaign (Only for external projects)
5. Automated User Satisfaction Analysis
6. Go Live Readiness Review
7. Go Live
8. Project Review & Effectiveness Evaluation

These changes prioritize efficiency by eliminating redundant tasks, automating manual ones, and ensuring steps are only taken when necessary, aligning with your organization's focus on optimizing processes and minimizing rework. Let me know if you'd like me to make additional refinements or if any step in particular should be explored further!

LLM-Output 40: Go-Live Process Optimization (Bitmap) R3

Given your organization's prioritization of efficiency, minimizing internal controls, and post-Go-Live feedback collection, I would refine the previous optimizations with a stronger emphasis on parallelizing tasks and removing steps that do not directly add stakeholder value.

Updated Process Optimizations:

1. **Parallelize "Final Test" and "Internal Approval"**
    - **Change:** Run the "Final Test" and "Internal Approval of successful deployment" steps in parallel, since both aim to confirm the deployment's success.
    - **Reason:** Parallel execution speeds up the process by reducing delays between validation and approval. This also fits your focus on reducing internal controls that don't add stakeholder value.
2. **Combine User Satisfaction Analysis with Go-Live**
    - **Change:** Remove "Manual User Satisfaction Analysis" as a separate step. Instead, gather user feedback during the "Go Live" phase through automated systems, and log it post-Go-Live for analysis.
    - **Reason:** As tickets are created post-Go-Live, feedback will naturally be collected during the live system phase. There's no need for a separate manual analysis step.
3. **Eliminate Project Presentation Unless Required**
    - **Change:** Remove the "Project Presentation to Management" step unless it is absolutely necessary based on the strategic importance of the project.

- **Reason:** Only projects that significantly impact stakeholders or management decisions should require a formal presentation, saving time and resources for both the project manager and management.

4. **Run PR Campaign in Parallel with Go-Live Readiness**

   - **Change:** Launch the "PR Campaign" preparation (if needed) in parallel with the "Go Live Readiness Review."
   - **Reason:** This reduces delays between PR preparation and system launch, aligning communication with technical execution.

5. **Integrated Go-Live and Ticket Management**

   - **Change:** Ensure that the "Go Live" step includes an automated system for logging user feedback and tickets, making feedback gathering and resolution part of the Go-Live phase itself.
   - **Reason:** This removes the need for a separate feedback phase and speeds up the response to any issues that arise after the system is live.

Final Optimized Process Flow:

- **Deployment on Production Systems**
- **Parallel: Final Test & Internal Approval**
- **Optional: Project Presentation to Management (only if strategically important)**
- **Parallel: PR Campaign & Go Live Readiness Review**
- **Go Live (includes feedback collection and ticket management)**
- **Project Review & Effectiveness Evaluation**

These changes reduce the overall process timeline by merging and parallelizing tasks where feasible, while maintaining the integrity of the go-live feedback and ticketing system, aligned with your organization's efficiency goals. Let me know if further adjustments are needed!

LLM-Output 41: Context-enhanced Go-Live Process Optimization (Bitmap) R3

R4

To optimize the process shown in the diagram, I suggest the following improvements to increase efficiency, reduce redundancy, and enhance clarity:

1. **Combine "Deployment on Production Systems" and "Internal Approval of successful deployment"**:

   - **Change**: Merge these two steps into a single stage called "Deployment and Internal Approval" to streamline the process and eliminate redundant approval stages. The internal approval can be integrated within the deployment workflow.
   - **Reason**: This reduces unnecessary approval steps and streamlines the deployment process.

2. **Final Test**:

   - **Change**: Implement a well-defined automated testing process within the "Final Test" step. Automated tests should run concurrently with deployment, reducing manual checks and speeding up the process.
   - **Reason**: Parallel testing and automation reduce waiting times and ensure faster identification of issues.

3. **Eliminate the "Project Presentation to Management" decision node**:

   - **Change**: Pre-schedule a management check-in earlier in the timeline instead of waiting until after testing. Only projects with special management requirements should require an additional presentation at this stage.

- **Reason**: This removes a potential bottleneck and streamlines management approvals earlier in the project.

4. **PR Campaign**:

   - **Change**: Allow the "PR Campaign" to be prepared concurrently with the "Final Test" step to avoid bottlenecks.
   - **Reason**: Running PR preparations in parallel ensures there are no delays, and the campaign is ready once the final test concludes.

5. **Manual User Satisfaction Analysis**:

   - **Change**: Replace the "Manual User Satisfaction Analysis" with an automated feedback collection system integrated into the live system.
   - **Reason**: Automation ensures consistent, real-time feedback collection and saves time, reducing manual effort.

6. **Go Live Step**:

   - **Change**: Introduce a "Go Live Checklist" to ensure all necessary steps and checks are completed before the system goes live.
   - **Reason**: A standardized checklist ensures that no critical steps are missed, enhancing the reliability of the Go-Live phase.

7. **Combine “Project Review & Effectiveness Evaluation” with Lessons Learned**:

   - **Change**: Merge the "Project Review" and "Lessons Learned" stages into a single feedback and evaluation step.
   - **Reason**: Combining these steps saves time and ensures a more efficient review process.

Optimized Process Summary:

1. **Deployment and Internal Approval** (Combined step)
2. **Final Test** (With parallel automated testing)
3. **Scheduled Management Check-In** (Pre-deployment)
4. **PR Campaign Preparation** (Runs in parallel with final test)
5. **Automated User Feedback Collection** (Replaces manual analysis)
6. **Go Live** (With standardized checklist)
7. **Combined Project Review & Lessons Learned**

These changes aim to improve process flow by minimizing delays, reducing manual steps, and encouraging automation and parallel activities, ensuring a more efficient and scalable system.

LLM-Output 42: Go-Live Process Optimization (Bitmap) R4

Based on your organization’s approach and focus on reducing unnecessary controls and parallelizing tasks, here are further optimizations to streamline the process:

Optimized Process (Considering Efficiency Focus)

1. **Deployment and Internal Approval Combined:**

   - Since internal approval often involves the same team that manages deployment, merge both tasks into a single step. Internal controls that don’t add value to stakeholders can be minimized, focusing on task completion in a single flow.

- **Change:** Deployment and approval are now handled in one efficient step with minimal internal checks.

2. **Final Test:**
   - The final test can be partially automated and run in parallel with other steps. For example, while the system is undergoing final tests, the PR campaign can begin preparing announcements or key updates, reducing the time gap between testing and going live.
   - **Change:** The "Final Test" is parallelized with "PR Campaign Preparation" to speed up execution.

3. **Remove Project Presentation to Management:**
   - Since your organization reduces team-internal controls and focuses on stakeholder value, this step can be eliminated unless absolutely necessary. Management presentations could be scheduled earlier or combined with other key decision points like the pre-Go-Live checklist.
   - **Change:** Remove the "Project Presentation to Management" step as it might not add immediate stakeholder value unless there is a major concern.

4. **PR Campaign in Parallel:**
   - The PR campaign can fully prepare and initiate during the testing phase. This ensures that the team is ready for communication once the system goes live, removing delays caused by waiting for post-test results.
   - **Change:** Run the "PR Campaign" entirely in parallel with the "Final Test."

5. **Automated Feedback Collection Post-Go-Live:**
   - As tickets are created post-Go-Live based on real-time feedback, the step “Manual User Satisfaction Analysis” should be replaced with an automated user feedback system. This feedback will generate tickets automatically, providing an efficient way to manage issues without manual intervention.
   - **Change:** Implement "Automated Feedback Collection" post-Go-Live.

6. **Streamlined Project Review & Lessons Learned:**
   - Combine project review and lessons learned into one streamlined session post-Go-Live, focusing on ticket data, feedback analysis, and key insights to adapt future processes. This removes redundant review steps.
   - **Change:** Combine "Project Review" with "Effectiveness Evaluation" and "Lessons Learned."

Optimized Process Flow:

- **Deployment and Internal Approval** (Combined step)
- **Final Test** (Run in parallel with PR Campaign)
- **PR Campaign** (Parallel with Final Test)
- **Go Live**
- **Automated User Feedback Collection** (Post-Go-Live tickets generated)
- **Combined Project Review, Effectiveness Evaluation, and Lessons Learned**

This further refinement aligns with your organizational goal to prioritize stakeholder value and reduce unnecessary internal controls, focusing on parallelizing tasks and minimizing dependencies.

LLM-Output 43: Context-enhanced Go-Live Process Optimization (Bitmap) R4

R5

To optimize this process, here are the key suggestions and changes made:

1. **Combine "Internal Approval of Successful Deployment" and "Final Test"**:
    - Currently, both steps involve post-deployment validation, which can be merged into a single “Final Validation & Testing” phase to reduce redundancy and save time.
2. **Automate the "Manual User Satisfaction Analysis"**:
    - If feasible, replacing the manual user analysis with an automated survey or live feedback mechanism would streamline this process. This could involve collecting data directly from users through the live system, reducing manual efforts.
3. **Parallelize PR Campaign with Project Presentation**:
    - The PR campaign and presentation to management are currently sequential. If these are independent of each other, running them in parallel will shorten the timeline. The PR campaign can be pre-planned based on initial approval and go-live readiness.
4. **Automate "Go Live" Announcements**:
    - Set up automated notifications or triggers for live system announcements and user communications, reducing the need for manual effort by the PR team.
5. **Review and Effectiveness Evaluation Merged with "Lessons Learned"**:
    - The "Project Review & Effectiveness Evaluation" and "Lessons Learned" are closely tied. Combine these steps to form a comprehensive review phase that simultaneously evaluates effectiveness and extracts lessons, ensuring continuous improvement.
6. **Conditional Logic for "Presentation to Management"**:
    - Streamline the decision-making process to determine earlier whether management presentation is needed. This could be done before the final test phase, reducing any delays due to awaiting presentation feedback.

By introducing automation where possible, merging similar steps, and running parallel processes, the process timeline is shortened and becomes more efficient, while still ensuring thorough validation and communication.

LLM-Output 44: Go-Live Process Optimization (Bitmap) R5

Based on your organization’s priorities and feedback mechanisms, here are additional tailored optimizations for your process:

1. **Eliminate "Manual User Satisfaction Analysis" Before Go-Live:**
    - Since feedback is gathered only post-Go-Live and tickets are created after the system is live, the "Manual User Satisfaction Analysis" step before Go-Live becomes redundant. Instead, integrate this into the post-Go-Live phase, which is already covered through live-system feedback and ticket creation.
2. **Merge "Go Live" and "User Satisfaction Analysis" Steps:**
    - Combine the "Go Live" phase with the process for collecting feedback. Since your organization gathers feedback post-Go-Live, streamline the transition from deployment to user feedback analysis within the live environment itself, reducing the need for a separate analysis step.
3. **Parallelize "Project Presentation to Management" with PR Campaign:**
    - If the same team members are involved in both management presentations and PR campaigns, these tasks can be run concurrently. This reduces delays due to sequential dependencies, speeding up the process flow.

4. **Eliminate or Automate Non-Essential Team-Internal Controls:**

    - If there are any steps or approval phases that are purely internal and don't add clear value to stakeholders (such as additional internal reviews), they should either be eliminated or automated to avoid unnecessary bottlenecks.

5. **Simplify Final Test and Approval:**

    - Combine the "Final Test" and "Internal Approval of Deployment" into one validation phase. This ensures the system is tested and approved in a single streamlined step, cutting down on back-and-forth between teams.

6. **Post-Go-Live Ticket Creation:**

    - Since tickets are generated post-Go-Live based on feedback, ensure that your "Go Live" process includes an automatic trigger for ticket collection and categorization. This can include deploying mechanisms to gather user feedback directly into the ticketing system, which can save time and reduce manual follow-ups.

By aligning the process with the organization's focus on efficiency and stakeholder value, these optimizations further simplify execution, avoid redundant checks, and help ensure parallel progress where feasible.

LLM-Output 45: Context-enhanced Go-Live Process Optimization (Bitmap) R5

Overview of Generative AI Tools Used

During the writing of this thesis, *GPT-4o* was used in various supporting functions:

- *Advisor*: Refining ideas, structuring content, evaluating data, and verifying calculations, particularly accuracy scores.
- *LaTeX Formatting*: Assistance in formatting tasks such as the creation of complex table formats.
- *BibTeX Formatting*: Support in formatting references to align with proper citation standards.
- *Formulations and Wording*: Assisting with alternative phrasing to improve clarity and maintaining a consistent scientific writing style, while also helping to optimize the document's layout.
- *Study Subject*: As the object of this study itself, to analyze the capabilities of LLMs for BPO.

Furthermore, we also leveraged *DeepL* for:

- *Formulations and Wording*: Assisting with alternative wording to enhance clarity, and supporting the translation of the *Abstract* into the German *Kurzfassung*, ensuring alignment and quality phrasing in both languages.

Despite the use of these ML-driven tools, all ideas, approaches, and the content of this work were produced independently by myself, with guidance supervisor. The models were solely used in a supportive and advisory role, similar to how other tools are used in the research process.

List of Figures

1.1 Timeline of BPM Automation Tool Emergence 2
1.2 State of research for LLM-driven BPM 3

2.1 The Demming (PDCA) Cycle 7
2.2 The BPM-Lifecycle and the focus of this work in it 8
2.3 Overview over LLM aspects covered in this section 10

3.1 Our Research Process 19

4.1 Main Process of the digitization process 25
4.2 Research Process detailed for the Evaluation Framework 29

9.1 Original Analysis Process 70
9.2 Adapted Analysis Process 71
9.3 Original Design Process 85
9.4 Adapted Design Process 86
9.5 Original Implementation Process 87
9.6 Adapted Implementation Process 88
9.7 Original Go-Live Process 89
9.8 Adapted Go-Live Process 90

List of Tables

5.1 Points Allocation for Recognition of 7PMG Violations 37
5.2 Points Allocation for Context-Enhanced Recognition of 7PMG Violations 40

6.1 Distribution of Optimization Opportunities Across Business Processes . . 45
6.2 Points Allocation for Found and Solved Opportunities: Design Process . . 48
6.3 Points Allocation for Found and Solved Opportunities: Implementation Process 49
6.4 Points Allocation for Found and Solved Opportunities: Go-Live Process . 50
6.5 Normalized Accuracy Scores by Process and Heuristic Application 51
6.6 Points Allocation for Found and Solved Opportunities: Design Process (Context-Enhanced) 56
6.7 Points Allocation for Found and Solved Opportunities: Implementation Process (Context-Enhanced) 57
6.8 Points Allocation for Found and Solved Opportunities: Go-Live Process (Context-Enhanced) 58
6.9 Normalized Accuracy Scores by Process and Heuristic Application (Context-Enhanced) 59

Acronyms

7PMG Seven Process Modeling Guidelines. 33–36, 38, 39

AR Action Research. 4, 5, 17–19, 21, 29, 43, 64

BPM Business Process Management. 1, 2, 7–9, 12–15, 20, 24, 33, 45, 63, 64, 67, 137

BPMN Business Process Model and Notation. 13, 23, 25–27, 47

BPO Business Process Optimization. 1–5, 7–9, 12, 14, 15, 17, 18, 20, 21, 23, 28–30, 33, 41, 43, 45, 61, 64, 66–68

GPT Generative Pre-Trained Transformers. 2, 10, 13–15, 27

LLM Large Language Model. 2–5, 7, 10–15, 17–21, 23, 24, 26–30, 33–39, 41, 43–47, 52, 53, 55, 59–68

OCR Optical Character Recognition. 46, 47, 61

SOTA State of the Art. 1–3, 5, 7, 10, 15, 26, 27, 46, 63–65, 67

Bibliography

[7PM10] Seven process modeling guidelines (7pmg). *Information and Software Technology*, 52(2):127–136, 2010.

[AAAHABH21] Younis Al-Anqoudi, Abdullah Al-Hamdani, Mohamed Al-Badawi, and R. Hedjam. Using machine learning in business process re-engineering. *Big Data Cogn. Comput.*, 5:61, 2021.

[Aal13] Wil Aalst. Aalst, w.m.p.: Business process management: a comprehensive survey. isrn softw. eng. 1-37. *ISRN Software Engineering*, pages ??–??, 01 2013.

[AI24a] Google AI. Gemini api documentation: Models. https://ai.google.dev/gemini-api/docs/models/gemini, 2024. Accessed: 2024-08-06.

[AI24b] Meta AI. Llama 3.1 model cards and prompt formats. https://llama.meta.com/docs/model-cards-and-prompt-formats/llama3_1, 2024. Accessed: 2024-08-06.

[Alt23] Steven A. Altman. The state of globalization in 2023. *Harvard Business Review*, 2023.

[ArX] ArXiv. Arxiv. https://arxiv.org/. Accessed: August 21, 2024.

[AVL20] Tahir Ahmad and Amy Van Looy. Business process management and digital innovations: A systematic literature review. *Sustainability*, 12(17), 2020.

[Bas99] Richard L Baskerville. Investigating information systems with action research. *Communications of the association for information systems*, 2(1):19, 1999.

[ber] *Evaluating Large Language Models in Process Mining: Capabilities, Benchmarks, and Evaluation Strategies.*

[BFI$^+$22] Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Noack, Hendrik Patzlaff, Felix Naumann, and Hazar Harmouch. The effects of data quality on machine learning performance, 2022.

[BGMMS21] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? *FAccT*, pages 610–623, 2021.

[BM04] Richard Baskerville and Michael D Myers. Special issue on action research in information systems: Making is research relevant to practice: Foreword. *MIS quarterly*, pages 329–335, 2004.

[BP97] A. Berg and P. Pottjewijd. *Workflow: Continuous Improvement by Integral Process Management.* Academic Service, Schoonhoven, 1997. [in Dutch].

[BPSM$^+$08] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml) 1.0 (fifth edition). `https://www.w3.org/TR/xml/`, November 2008. W3C Recommendation.

[Buz96] John A. Buzacott. Commonalities in reengineered business processes: Models and issues. *Management Science*, 42(5):768–782, 1996.

[BWH98] Richard Baskerville and A Trevor Wood-Harper. Diversity in information systems action research methods. *European Journal of information systems*, 7:90–107, 1998.

[Cai24] Wendy Cain. Prompting change: Exploring prompt engineering in large language model ai and its potential to transform education. *TechTrends*, 68:47–57, 2024.

[Cel24] Celonis SE. Celonis: Execution management system. `https://www.celonis.com/`, 2024. Accessed: 2024-08-14.

[CGM09] Mike Chiasson, Matt Germonprez, and Lars Mathiassen. Pluralist action research: a review of the information systems literature. *Information systems journal*, 19(1):31–54, 2009.

[Cla24] Claude 3.5: A sonnet for ai progress. Anthropic News, 2024. `https://www.anthropic.com/news/claude-3-5-sonnet`.

[CNB22] CNBC. Electronics set to rise as chip giants like tsmc, samsung hike prices. *CNBC*, 2022. Accessed: 2024-08-09.

[Com21] McKinsey & Company. The potential of machine learning in services operations. 2021.

[Con10] Sue Conger. *Six Sigma and Business Process Management*, pages 127–148. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[CZLZ24] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review, 2024.

[DRMR13] M. Dumas, M. Rosa, J. Mendling, and H. Reijers. Introduction to business process management. pages 1–31, 2013.

[ETdROR24] Bedilia Estrada-Torres, Adela del Río-Ortega, and Manuel Resinas. Mapping the landscape: Exploring large language model applications in business process management. In Han van der Aa, Dominik Bork, Rainer Schmidt, and Arnon Sturm, editors, *Enterprise, Business-Process and Information Systems Modeling*, pages 22–31, Cham, 2024. Springer Nature Switzerland.

[For24] World Economic Forum. 9 forces reshaping the global business landscape. `https://www.weforum.org/agenda/2024/01/9-forces-reshaping-the-global-business-landscape/`, 2024. Accessed: 2024-08-09.

[GAER23] Michael Grohs, Luka Abb, Nourhan Elsayed, and Jana-Rebecca Rehse. Large language models can accomplish business process management tasks, 2023.

[HC93] Michael Hammer and James Champy. *Reengineering the Corporation: A Manifesto for Business Revolution.* Harper Business Editions, New York, 1993.

[HC23] Arash Hajikhani and Carolyn Cole. A critical review of large language models: Sensitivity, bias, and the path toward specialized ai. =https://arxiv.org/abs/2307.15425, 2023.

[HJSM24] Maeve Hutchinson, Radu Jianu, Aidan Slingsby, and Pranava Madhyastha. Llm-assisted visual analytics: Opportunities and challenges, 2024.

[HMPR04] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, pages 75–105, 2004.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[HSBQ24] Viviane Herdel, Sanja Sćepanović, Edyta Bogucka, and Daniele Quercia. Exploregen: Large language models for envisioning the uses and risks of ai technologies. =https://arxiv.org/abs/2407.12454, 2024.

[HW98] H. Hungenberg and Torsten Wulf. Business process reengineering. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 93:304 – 304, 1998.

[JLF+23] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, March 2023.

[JYS+24] Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. The impact of reasoning step length on large language models, 2024.

[KBSvdA24] Humam Kourani, Alessandro Berti, Daniel Schuster, and Wil M. P. van der Aalst. Process modeling with large language models, 2024.

[KHH18] Tatic Kasim, Mahir Haracic, and Merima Haracic. The improvement of business efficiency through business process management. *Economic Review: Journal of Economics and Business*, 16(1):31–43, 2018.

[Kir22] Mathias Kirchmer. Agile innovation through business process management: Realizing the potential of digital transformation. In Boris Shishkov, editor, *Business Modeling and Software Design*, pages 21–34, Cham, 2022. Springer International Publishing.

[Kle95] Michael Klein. 10 principles of reengineering. *Executive Excellence*, 12(2):20, 1995.

[KMN24] Alex Kim, Maximilian Muhn, and Valeri Nikolaev. Financial statement analysis with large language models. =https://arxiv.org/abs/2407.17866, 2024.

[KWR+23] Timotheus Kampik, Christian Warmuth, Adrian Rebmann, Ron Agam, Lukas N. P. Egger, Andreas Gerber, Johannes Hoffart, Jonas Kolk, Philipp Herzig, Gero Decker, Han van der Aa, Artem Polyvyanyy, Stefanie Rinderle-Ma, Ingo Weber, and Matthias Weidlich. Large process models: Business process management in the age of generative ai, 2023.

[Lew46] Kurt Lewin. Action research and minority problems. *Journal of social issues*, 2(4):34–46, 1946.

[Lie98] D. Liebeskind. Reengineering r&d work processes. *Research-technology Management*, 41:43–48, 1998.

[LMZ+23] Beibin Li, Konstantina Mellou, Bo Zhang, Jeevan Pathuri, and Ishai Menache. Large language models for supply chain optimization, 2023.

[LNA^{+}23] Luis Fernando Lins, Nathalia Nascimento, Paulo Alencar, Toacy Oliveira, and Donald Cowan. Comparing generative chatbots based on process requirements, 2023.

[LXZ23] Qingquan Liu, Chao Xie, and Jie Zhou. Pre-trained language models: The next big thing in artificial intelligence. *Journal of Artificial Intelligence*, 50:75–98, 2023.

[LZY^{+}24] Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need, 2024.

[LZZ^{+}24] Haiyun Li, Qihuang Zhong, Ke Zhu, Juhua Liu, Bo Du, and Dacheng Tao. Iterative data augmentation with large language models for aspect-based sentiment analysis. =https://arxiv.org/abs/2407.00341, 2024.

[MFK^{+}21] Niels Martin, Dominik A. Fischer, Georgi D. Kerpedzhiev, et al. Opportunities and challenges for process mining in organizations: Results of a delphi study. *Business & Information Systems Engineering*, 63(5):511–527, 2021.

[Mic23a] Microsoft Learn. Microsoft copilot for microsoft 365 overview. https://learn.microsoft.com/en-us/microsoft-365/copilot/overview, 2023. Accessed: 2024-08-01.

[Mic23b] Microsoft News Center. Introducing microsoft 365 copilot: your copilot for work. https://news.microsoft.com/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work/, 2023. Accessed: 2024-08-01.

[Mic24a] Microsoft. Advanced prompt engineering. https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/advanced-prompt-engineering, 2024. Accessed: 2024-10-03.

[Mic24b] Microsoft 365. Microsoft 365, 2024. https://www.microsoft.com/en/microsoft-365?market=af.

[MLB^{+}16] James Manyika, Susan Lund, Jacques Bughin, Jonathan Woetzel, Kalin Stamenov, and Dhruv Dhingra. Digital globalization: The new era of global flows, 2016.

[MNvdA07] Jan Mendling, Gustaf Neumann, and Wil M.P. van der Aalst. Understanding the occurrence of errors in process models based on metrics. In Robert Meersman and Zahir Tari, editors, *OTM Conferences 2007, Proceedings, Part I*, volume 4803 of *Lecture Notes in Computer Science*, pages 113–130. Springer, 2007.

[MRC07] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What makes process models understandable? In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007, Proceedings*, volume 4714 of *Lecture Notes in Computer Science*, pages 48–63, Brisbane, Australia, 2007. Springer.

[MRR] Jan Mendling, H.A. Reijers, and Jan Recker. Activity labeling in process modeling: Empirical insights and recommendations. *Information Systems*. URL: `http://eprints.qut.edu.au/19625/`.

[MVvD+08] Jan Mendling, Hajo Verbeek, Boudewijn van Dongen, Wil M.P. van der Aalst, and Gustaf Neumann. Detection and prediction of errors in EPCs of the SAP reference model. *Data & Knowledge Engineering*, 64(1):312–329, 2008.

[MW23] Priyanka Mudgal and Rita Wouhaybi. An assessment of chatgpt on log data, 2023.

[New24a] Bloomberg News. Ai energy crisis boosts need for more efficient chips. *Bloomberg*, 2024. Accessed: 2024-08-09.

[New24b] Bloomberg News. Ai is key factor for chip stocks looking to extend record highs. *Bloomberg*, 2024. Accessed: 2024-08-09.

[New24c] Bloomberg News. Nvidia is world's most valuable company. `https://www.bloomberg.com/news/newsletters/2024-06-18/bloomberg-evening-briefing-nvidia-is-world-s-most-valuable-com` 2024. Accessed: 2024-08-18.

[New24d] Bloomberg News. Tsmc sales grow 45% in july on strong ai chip demand. *Bloomberg*, 2024. Accessed: 2024-08-09.

[NLZ+23] Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. Can generalist foundation models outcompete special-purpose tuning? case study in medicine, 2023.

[NP20] Miroslava Nyulásziová and Dana Paľová. Implementing a decision support system in the transport process management of a small slovak transport company. *Journal of Entrepreneurship, Management and Innovation*, 2020.

[Ope18] OpenAI. Improving language understanding by generative pre-training. `https://openai.com/index/language-unsupervised/`, 2018. Accessed: 2024-08-06.

[Ope21] OpenAI. Multimodal neurons in artificial neural networks. https://openai.com/index/multimodal-neurons/, 2021.

[Ope23a] OpenAI. Clip: Connecting text and images. https://openai.com/research/clip-connecting-text-and-images/, 2023.

[Ope23b] OpenAI. Gpt-4. https://openai.com/index/gpt-4-research/, 2023. Accessed: 2024-08-09.

[Ope23c] OpenAI. Gpt-4v(ision) system card. https://openai.com/research/gpt-4v-system-card/, 2023.

[Ope24a] OpenAI. Chatgpt: Your ai assistant. https://openai.com/chatgpt/, 2024. Accessed: 2024-08-06.

[Ope24b] OpenAI. Gpt-3.5 turbo documentation. https://platform.openai.com/docs/models/gpt-3-5-turbo, 2024. Accessed: 2024-08-06.

[Ope24c] OpenAI. Introducing gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024. Accessed: 2024-08-06.

[Ope24d] OpenAI. Prompt engineering guide. https://platform.openai.com/docs/guides/prompt-engineering/prompt-engineering, 2024.

[Pet99] Peter B. Petersen. Total quality management and the deming approach to quality management. *Journal of Management History (Archive)*, 5(8):468–488, 1999.

[POS12] Nikolaos Papas, Robert M O'keefe, and Philip Seltsikas. The action research vs design science debate: reflections from an intervention in egovernment. *European Journal of Information Systems*, 21(2):147–159, 2012.

[PR95] Joe Peppard and Philip Rowland. *The Essence of Business Process Reengineering.* Prentice-Hall Editions, New York, 1995.

[Red05] Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega*, 33(4):283–306, 2005.

[Res] ResearchGate. Researchgate. https://www.researchgate.net/. Accessed: August 21, 2024.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[RM08] Jan Recker and Jan Mendling. On the translation between bpmn and bpel: Conceptual mismatch between process modeling languages. In T. Latour and M. Petit, editors, *Proceedings of the CAiSE Workshops at the 18th Conference on Advanced Information Systems Engineering*, 2008.

[RR94] Ronald O. Rupp and J. R. Russell. The golden rules of process redesign. *Quality Progress*, 27(12):85–92, 1994.

[RvB15] Michael Rosemann and Jan vom Brocke. *The Six Core Elements of Business Process Management*, pages 105–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[S$^+$22] Victor Sanh et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2022.

[SBW24] Marek Szelągowski and Justyna Berniak-Woźny. Bpm challenges, limitations and future development directions – a systematic literature review. *Business Process Management Journal*, 30(2):505–557, 2024.

[SGM19] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, 2019.

[Sof24] Software AG. Aris: Business process management platform. `https://www.aris.com/`, 2024. Accessed: 2024-08-14.

[Spr] SpringerOpen. Springeropen. `https://www.springeropen.com/`. Accessed: August 21, 2024.

[SS97] Abraham Seidmann and Arun Sundararajan. The effects of task and information asymmetry on business process redesign. *International Journal of Production Economics*, 50(2-3):117–128, 1997.

[Tim24] Financial Times. Artificial intelligence faces regulation challenges in europe. `https://www.ft.com/content/96dfec5f-4d5f-4c3e-8f66-ebd0dfc8392d`, July 2024. Accessed: 2024-08-18.

[VBM23a] Maxim Vidgof, Stefan Bachhofner, and Jan Mendling. Large language models for business process management: Opportunities and challenges. In Chiara Di Francescomarino, Andrea Burattin, Christian Janiesch, and Shazia Sadiq, editors, *Business Process Management Forum*, pages 107–123, Cham, 2023. Springer Nature Switzerland.

[VBM23b] Maxim Vidgof, Stefan Bachhofner, and Jan Mendling. Large language models for business process management: Opportunities and challenges. In Chiara Di Francescomarino, Andrea Burattin, Christian Janiesch, and Shazia Sadiq, editors, *Business Process Management Forum*, pages 107–123, Cham, 2023. Springer Nature Switzerland.

[vdA11] Wil M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer, 2011.

[vdAtHKB03] Wil van der Aalst, A. ter Hofstede, B. Kiepuszewski, and A. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.

[VdAVH02] Wil MP Van der Aalst and Kees M. Van Hee. *Workflow Management: Models, Methods, and Systems.* MIT Press Editions, Cambridge, 2002.

[vdAW21] Wil M.P. van der Aalst and M. Wil. Hybrid intelligence : to automate or not to automate, that is the question. *International Journal of Information Systems and Project Management*, 9:5–20, 2021.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[W+22] Jason Wei et al. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[WH24] Sean Williams and James Huckle. Easy problems that llms get wrong, 2024.

[WIL+23] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. =https://arxiv.org/abs/2303.17564, 2023.

[WLC17] Leslie Willcocks, Mary Lacity, and Andrew Craig. Robotic process automation: strategic transformation lever for global business services? *Journal of Information Technology Teaching Cases*, 7(1):17–28, 2017.

[WXJ+23] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023.

[YWL+24] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers, 2024.

[ZMLA24] Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. Enhancing contextual understanding in large language models through contrastive decoding. =https://arxiv.org/abs/2405.02750, 2024.

[ZZPC23] Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. Context-faithful prompting for large language models. =https://arxiv.org/abs/2303.11315, 2023.